CIM Series in Mathematical Sciences 4

João Paulo Almeida
José Fernando Oliveira
Alberto Adrego Pinto *Editors*

Operational Research

IO 2013 - XVI Congress of APDIO, Bragança, Portugal, June 3–5, 2013

Springer

CIM Series in Mathematical Sciences

Volume 4

Series Editors:
Irene Fonseca
Department of Mathematical Sciences
Center for Nonlinear Analysis
Carnegie Mellon University
Pittsburgh, PA, USA

Alberto Adrego Pinto
Department of Mathematics
University of Porto, Faculty of Sciences
Porto, Portugal

The CIM Series in Mathematical Sciences is published on behalf of and in collaboration with the Centro Internacional de Matemática (CIM) in Coimbra, Portugal. Proceedings, lecture course material from summer schools and research monographs will be included in the new series.

More information about this series at
http://www.springer.com/series/11745

João Paulo Almeida • José Fernando Oliveira • Alberto Adrego Pinto
Editors

Operational Research

IO 2013 - XVI Congress of APDIO, Bragança, Portugal, June 3–5, 2013

Editors
João Paulo Almeida
Department of Mathematics and
LIAAD-INESC TEC
Polytechnic Institute of Bragança
School of Technology and Management
Bragança, Portugal

José Fernando Oliveira
Faculty of Engineering
CEGI - INESC TEC
University of Porto
Porto, Portugal

Alberto Adrego Pinto
Department of Mathematics and
LIAAD-INESC TEC
University of Porto
Porto, Portugal

ISSN 2364-950X
CIM Series in Mathematical Sciences
ISBN 978-3-319-20327-0
DOI 10.1007/978-3-319-20328-7

ISSN 2364-9518 (electronic)

ISBN 978-3-319-20328-7 (eBook)

Library of Congress Control Number: 2015950648

Mathematics Subject Classification (2010): 90Bxx

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Foreword

The main objectives of the APDIO – the Portuguese OR Society (in Portuguese: Associação Portuguesa de Investigação Operacional) are to disseminate the latest advances in Operational Research, its best practices, and, furthermore, to foster the bonds within the OR Community, helping it to pursue its research interests and meet future challenges. Accordingly, the APDIO promotes Operational Research through courses, seminars, workshops, and conferences, while also providing editorial support for scientific publications (e.g., scientific journals, newsletters, and books). The APDIO is a scientific society that brings together Portugal's Operational Research Community. It was created in 1978 by 140 founding members, including university researchers, industrial practitioners, and several Portuguese institutes and companies as institutional associates. The APDIO has been a member of the International Federation of Operational Research Societies (IFORS) and International Federation of Automatic Control (IFAC) since its inception. It has gone on to become a member of the Association of European Operational Research Societies (EURO) and was involved in the creation of the Association of Latin-Iberoamerican Operational Research Societies (ALIO) in 1982. Over the years, the APDIO has organized a total of 16 National Conferences: Lisbon, 1982; Porto, 1984; Coimbra, 1987; Lisbon, 1989; Évora, 1992; Braga, 1994; Aveiro, 1996; Faro, 1998; Setúbal, 2000; Guimarães, 2002; Porto, 2004; Lisbon, 2006; Vila Real, 2008; Caparica, 2009; Coimbra, 2011; and Bragança, 2013. The next National Conference will be held in Portalegre in September 2015.

The publication of this volume, with a selection of papers from IO2013 – the 16th National Conference of the APDIO, held in Bragança, Portugal, June 3–5, 2013 – is in keeping with the society's main purposes. We hope that it will be the first volume in a Springer Edition Series devoted to the main findings presented at our National Conferences.

We are very much indebted to the Editors of this volume, Professors João Paulo Almeida, José Fernando Oliveira, and Alberto Adrego Pinto, to whom I express my gratitude for having embraced this project and brought it to fruition. I am also grateful to the authors who contributed to this volume; their papers are excellent examples of the current research activities of the Portuguese OR Community

members. Lastly, I would also like to thank the reviewers, whose anonymous work was essential to guaranteeing the publication's high quality.

Aveiro, Portugal
May 25, 2015

Domingos Moreira Cardoso
(President of the Directive Committee of the APDIO)

Acknowledgments

We thank all the authors for their contributed chapters and all the anonymous referees.

We thank the president of APDIO – Associação Portuguesa de Investigação Operacional – Professor Domingos Moreira Cardoso.

We thank the Executive Editor for Mathematics, Computational Science, and Engineering at Springer-Verlag, Martin Peters, for invaluable suggestions and advice and Ruth Allewelt at Springer-Verlag for assistance throughout this project.

João Paulo Almeida and Alberto Adrego Pinto would like to thank LIAAD-INESC TEC and gratefully acknowledge the financial support received by the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) – within project UID/EEA/50014/2013 and ERDF (European Regional Development Fund) through the COMPETE Program (operational program for competitiveness) and by National Funds through the FCT within Project "Dynamics, optimization and modelling" with reference PTDC/MAT-NAN/6890/2014. Alberto Adrego Pinto also acknowledges the financial support received through the *Special Visiting Researcher* scholarship program at IMPA, Brazil.

Bragança, Portugal
Porto, Portugal

João Paulo Almeida
José Fernando Oliveira
Alberto Adrego Pinto

jpa@ipb.pt

Contents

Performance Evaluation of Parfois Retailing Stores

Maria Emília Dias Alves and Maria C.A. Silva Portela

Abstract This study describes a method for the assessment of retail store performance using Data Envelopment Analysis (DEA). The method is applied to the Portuguese company, Parfois, operating in retail fashion accessories, where we analyze the efficiency of 63 stores operating in Portugal. Firstly, we present a literature review on the subject, followed by a description of the company and the DEA method applied to the Parfois stores. Then, some of the factors potentially affecting the efficiency of Parfois stores are also analyzed, and the implications of using this method are discussed. The main intention of the study is to show how DEA can be used as a support tool to the management of Parfois stores at a national level and to help the company, through the identification of areas of potential growth, to define objectives, to increase its performance and achieve excellence.

1 Introduction

Companies that manage a large number of stores, such as bank branches, travel agencies, supermarkets, or other sales outlets typically assess their performance through productivity and financial ratios. Based on these ratios periodic performance targets are established. Ratios, however, have several limitations. Firstly they cannot take into account trade-offs that may happen between factors (e.g. a high level of productivity per worker, may not be compatible with large service times, which may be perceived by customers as a dimension of the quality of the service provided), and secondly they only take into account one dimension of performance at a time (for details see [5]). Data Envelopment Analysis (DEA) is a method that can overcome some of the limitations of performance analysis based on ratios, and therefore is a good alternative for assessing the performance of retail stores. Several authors

M.E.D. Alves (✉)
Parfois, Rua do Sistelo 755, 4435-429 Rio Tinto, Portugal
e-mail: Maria.Alves@parfois.com

M.C.A.S. Portela
Faculdade de Economia e Gestão, Universidade Católica no Porto, Rua Diogo Botelho 1327, 4169-005 Porto, Portugal
e-mail: csilva@porto.ucp.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_1

have applied this technique to retail stores, since the first application by Norman and Stocker [12]. Most existing retail applications have focused on the food retail sector, but there are some examples of non-food retail stores analysis, as we will see in Sect. 3.

In this paper we contribute to expand the literature on the non-food retail applications, through an application of DEA to Parfois retail stores. We assess the technical efficiency of a set of 63 stores (using data from 2011). The applied model incorporates restrictions on the weights assigned to each output, as we wanted to reflect in the DEA model the importance of each item for the company's sales as a whole. To the authors' knowledge there is only one study in the literature of retail stores' efficiency assessment that has applied weight constraints to the DEA models: The study of Thomas et al. [14] who assessed 520 furniture stores through a DEA weight restricted model.

In addition, we also analyse the impact of some factors, such as the location of the store, the concept of the store, or the quality of service on the efficiency of the stores. Our conclusions point for some inefficiencies identified in the Parfois stores, with some output items having a greater potential for improvement than others. In addition we conclude that the measure of service quality used by the store seems to be unrelated with efficiency, and this should be a matter of concern for the management of this organization.

This paper is organized as follows. In the next section we present the DEA model used in the assessment, and in Sect. 3 we present a brief review of the previous studies which applied the DEA technique to evaluate the efficiency of retail stores. Section 4 provides a brief description of the company used as a case study and the inputs and outputs selected. Then, the results obtained are discussed in Sects. 5 and 6 concludes.

2 Data Envelopment Analysis

The methodology employed to analyse the performance of Parfois retail stores was DEA. Consider for each Decision Making Unit (DMU) j ($j = 1, \ldots, n$) a vector $\mathbf{x_j} = (\mathbf{x_{1j}}, \mathbf{x_{2j}}, \ldots, \mathbf{x_{mj}})$ reflecting m inputs consumed for producing a vector of s outputs $\mathbf{y_j} = (\mathbf{y_{1j}}, \mathbf{y_{2j}}, \ldots \mathbf{y_{sj}})$.

The technical efficiency for unit o is obtained from the solution of model (1) (see e.g. Charnes, Cooper and Rhodes [6]), as the inverse of β, where constant returns to scale (CRS) are assumed.

$$\max_{\lambda_j, \beta} \left\{ \beta \mid \sum_{j=1}^{n} \lambda_j x_{ij} \leq x_{io},\ i = 1, \ldots, m,\ \sum_{j=1}^{n} \lambda_j y_{rj} \geq \beta y_{ro},\ r = 1, \ldots, s,\ \lambda_j \geq 0 \right\} \tag{1}$$

Note that model (1) is defined in the envelopment form of DEA models, The weights formulation is dual to the envelopment formulation and considers a different perspective of efficiency: that efficiency is obtained from a relative productivity measure defined as the ratio between the weighted sum of outputs and the weighted sum of inputs (or in fact the inverse of this ratio when an output oriented model is used). The weight's formulation for the output orientation is shown in model (2), where u_r and v_i, the weights assigned to outputs and inputs, respectively, are the decision variables.

$$\min_{v_i,u_r} \Big\{ \sum_{i=1}^{m} v_i x_{io} \mid \sum_{i=1}^{m} v_i x_{ij} - \sum_{r=1}^{s} u_r y_{rj} \geq 0,\ j = 1, \ldots, n, \quad \sum_{r=1}^{s} u_r y_{ro} = 1, \quad v_i, u_r \geq 0 \Big\} \tag{2}$$

The weights formulation is usually employed when additional restrictions on factor weights are imposed. This may be justified by several reasons: (i) the need to improve discrimination between efficient units; (ii) the need to avoid the placement of zero weights on some factors; (iii) the need to capture certain relationships between inputs and outputs, etc. (see for details, [13]). In our application to retail stores weight constraints were needed to avoid allowing units placing zero weights on some outputs. The type of weight constraints that were added to model (1) were assurance regions of type I, which take the form in (3), where α and β are user specified constants (see for details, [13]) and the indices k and p indicate one of the outputs r. The choice for AR type restrictions is related to the fact that these behave better than other type of constraints, that may cause unfeasibility to the model (like absolute weight restrictions), or may cause non-frontier units to act as benchmarks (as the case of virtual weight restrictions).

$$\alpha \leq \frac{u_k}{u_p} \leq \beta \tag{3}$$

3 Previous Studies on Retail Stores

The number of studies applying DEA to retail stores are not many, particularly when we consider non-food retail. In this section we present some examples of studies that influenced our modelling choices, focusing on the type of stores analysed, on the set of inputs and outputs used, on the evaluation methodology applied and on the inclusion of non-discretionary factors

The study of Norman and Stocker [12] was the first study to apply DEA to the retail sector. Most existing studies applied DEA to food-retail stores, like [11] who analyzed 25 Finnish supermarkets, Vaz [15], Vaz et al. [17] or Barros and Alves [4] who analyzed Portuguese supermarkets, Korhonen and Syrjänen [9] who evaluated the efficiency of 13 supermarkets, and Athanassopoulos and Ballantine [1], who evaluated the efficiency of several supermarket chains in the UK. Some studies

applied DEA to the non-food retail, like Grewal et al. [8], who evaluated the efficiency of 59 stores selling car components, and Thomas et al. [14] who developed a model to evaluate the efficiency of a furniture chain constituted by 520 stores. Köksal and Aksu [10] measured and compared the technical efficiency of 24 travel agencies in Turkey.

Although DEA has been the base methodology in the above studies, the modeling choices have been different. For example, Norman and Stocker [12] used three distinct models to evaluate the efficiency of 45 retail stores through different perspectives of store management: Cost efficiency, Market efficiency, and Revenue Efficiency, but traditional DEA models have been applied. Another example of the use of several models/perspectives of evaluation is that of Köksal and Aksu [9], who evaluated the efficiency of 13 supermarkets during 10 years through a 3 stage model.

Regarding modelling choices, some authors have used more complex DEA models to allow the re-allocation of some inputs within the stores (like in [4, 15, 17] and [11]) or between stores [1]. Vaz [15] used DEA to evaluate the performance of 70 stores of a Portuguese retail company including hypermarkets, supermarkets and small supermarkets (see also [17]). Each of these store layouts have different sets of products, different merchandising areas, different prices, and are usually located differently too. Each store was considered as a set of sections (meat, fruit and vegetables, hygiene articles, etc.) with autonomous management, and these sections were compared between stores rather than the whole store. For comparing the whole store, Vaz [15] and Vaz et al. [17] used the network DEA model of Färe et al. [7] for identifying objectives for each section, and for allowing re-allocation of resources between sections, that were compatible with the objective of the store as a whole (the maximisation of sales). Korhonen and Syrjänen [11] also used re-allocation models and applied them to a chain of Finnish supermarkets. Their objective was to define the allocation of total resources of the company to its 25 supermarkets, in a way that the total resources allocated (number of working hours, and total area) were constraint by a certain threshold.

Some authors have also used Malmquist indices to analyse the evolution of efficiency over time (e.g. [4] who assessed 47 supermarkets for the years of 1999 and 2000) or to compare different groups of stores (e.g. [16] compared supermarket and hypermarket's performance using a Malmquist type-index).

Most of existing studies of retail stores use models that seek output expansion rather than input contraction. Outputs are in general related to sales, where sales can be taken in value or in quantity and aggregated or disaggregated. For example, Banker et al. [3] considered aggregate sales (see also [9] or [1]), and [8] considered sales disaggregated by category of product as a way to understand the role of product diversity in the performance of the stores (see also [2]). The inputs considered in most of existing applications are similar. Generally there is an input associated to workforce, which can be the number of hours of work (as in [3]), number of full time or full time equivalent workers (as in [14] or [4]), or the costs with the workforce (as in [9] or [4] who considered both the number and cost of workforce as inputs of the model); another associated to the area of the store (e.g. [15, 17] or [3]); and another input associated to the average stock available for selling (e.g. [15, 17], and [3]).

Less conventional inputs relate to the value of damaged products used in Vaz [15], and Vaz et al. [17] (this is considered an undesirable output and treated within the DEA models as an input), or in Barros and Alves [4] an input relating to the number of teller machines.

When considering the set of inputs to include in the DEA models to assess retail stores' efficiency, it is common to consider a set of non-discretionary factors that also affect efficiency, like the age of the store, its location, the dimension of the population that it serves, the competition in the area, etc. Some authors have considered these factors as additional inputs of the DEA model, while others have considered them in second stage analysis. Examples of the former type of approach can be found in Thomas et al. [14], Grewal et al. [8] and Norman and Stoker [12]. Examples of the latter type of approach can be found in Ket and Chu (2003) [9], or Banker et al. [3]. Ket and Chu [9] compared efficiency scores with environmental variables to conclude that the bad performance of some stores can indeed be attributed to the environment where they are located, whereas Banker et al. [3] used a more sophisticated logarithmic regression model to analyse the impact of environmental factors like location, average family income of the population in the area, number of competitors in the area, number of supervisors per salesman, amongst others.

In our application to 63 Parfois stores, we took into account the existing literature on the subject regarding the choice of the input and output factors, the orientation of the models, the disaggregation of outputs into the categories of products sold, and incorporating non-discretionary factors into the analysis, like age, and area of the store, which were considered to impact directly on the sales values. Regarding other factors, such as location, store concept and the service quality, their relationship with efficiency was evaluated a posteriori.

4 The Company and the Variables Used

Parfois is a Portuguese company that was founded in 1994 and is currently owned by the group Barhold, SGPS, S.A. The company specializes in fashion accessories, and is responsible for the design and sales a great variety of fashion products from textile accessories, non-textile accessories, hair accessories, bijou, handbags, shoes and travel items. The company operates mainly through own stores in Portugal, Spain, France and Poland, and through franchised stores in other countries (Parfois operates in a total of 31 countries). At the end of 2011, Parfois had 109 stores in Portugal (68 of which are own stores), and owned 31 stores in Spain, 3 in France and 27 in Poland. The company employs about 1000 employees, and at the end of 2011 the sales of the group Barhold, SGPS, S.A. reached 58 million euros.

The aim of this paper is to report a evaluation exercise for 63 Parfois stores situated in Portugal, from which 16 are located in traditional commercial streets and the remaining are located in shopping centers. The evaluation exercise uses data from the year 2011. In selecting the set of stores to analyse we were careful to

Table 1 Inputs and Outputs used

Inputs	Outputs
Area of the store	Sales of textiles
Age	Sales of non-textiles
FTE	Sales hair products
Average Stock	Sales of bijou
	Sales of Shoes
	Sales of Handbags
	Sales of party bags
	Sales of evening bags
	Sales of wallets
	Sales of travel bags

include stores that were homogeneous and could be compared (e.g., stores located in airports or stores where some of the data were not available were excluded from the analysis). The set of inputs and outputs used in the technical efficiency of Parfois retail stores is shown in Table 1. The inputs considered include discretionary and non-discretionary factors. Discretionary inputs include full time equivalent personnel (FTE) and the average value of stock (these inputs are both indicative of the dimension of the stores and of their sales' volume). The non-discretionary factors considered were the age and area of the store, which are believed to influence the performance and sales volume of the stores. The area of the store cannot be easily changed in the short run, and therefore it was considered a non-discretionary factor. Note however that this distinction between inputs does not imply a different treatment of these in the model, as we used a output oriented model where all factors are by default treated as non-discretionary.

FTE is a factor that clearly determine sales, as the correlation between this input and an aggregate value of sales is 0.908 (p-value = 0.000), meaning that the relationship is positive and very strong. The average stock was considered in value (rather than in quantity) by multiplying the quantity of each type of stock by its average price in 2011, and then aggregating all types of stocks into a single aggregate value. The relationship between average stock and sales is positive and the correlation coefficient is 0.766.

Regarding non-discretionary inputs, the bigger the area of the store (measured in square meters) the higher the number of items that can be exposed to the clients, and the higher the number of clients and sales staff that can be within the store. The correlation coefficient between the area of the stores and the aggregate value of sales is 0.427 (statistically significant – p-value = 0.000). In a regression analysis we also verified that area and number of FTE were two important determinants of sales.

The age of the store (measured in months since opening date) is also a factor that showed up as statistically significant in the aggregate sales regression. This factor appears however, more relevant in street stores than in shopping center's stores. Figure 1 shows the relationship between age and sales (in thousand of Euros), being

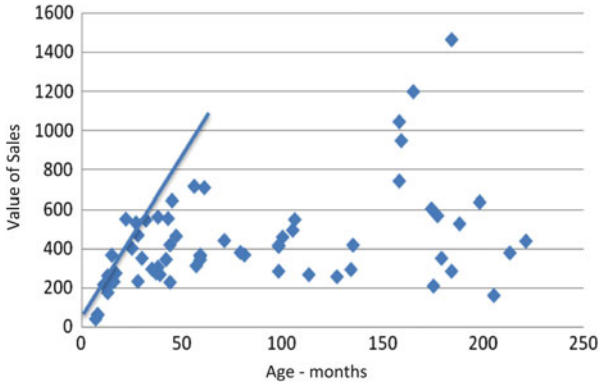


Fig. 1 Relationship between sales and age

evident that the trend for increasing sales with age is positive at least up to a certain point.

As far as outputs are concerned the objective is to reflect the activity of the stores, and therefore we considered the sales disaggregated by the various types of products in a store. All outputs have been considered in terms of the quantity of products sold and not in terms of its value, although the average price of each type of product varies. This means that ideally this technical efficiency analysis should be followed by a revenue efficiency analysis (but we do not pursue that avenue here).

5 Technical Efficiency Results

Efficiency results for the set of 63 stores were obtained from the application of model (1). Constant Returns to Scale (CRS) were used, since a preliminary analysis showed that using a Variable Returns to Scale (VRS) model would mainly benefit small stores. In addition we observed efficient CRS stores for all dimensions in our sample (as measured through area per FTE). Efficiency results showed about 28 stores without any potential for increasing their sales (efficient stores). The average efficiency was 89.37 %. In analysing these results, it was felt that the freedom given to stores in weighting their outputs lead to inadmissible weighting schemes. For example, some stores could weight mainly the sales of travel products, when its importance to the overall sales of the company is very small. Therefore it was important to impose constraints that could reflect the importance of each of the items on the output set. This importance could be seen in terms of volume or in terms of value. We computed both, but choose to focus our weight constraints on the value of the products since this is the criterion most valued by management. The information on the percentage of total sales (in value) of each type of item is shown in Table 2.

Table 2 Percentage of total sales per type of product

Product	Percentage
Textile products	8.80 %
Non-textile products	9.00 %
Hair products	2.00 %
Bijou	25.50 %
Shoes	8.10 %
Handbags	32.10 %
Party bags	1.60 %
Evening bags	3.30 %
Wallets	8.60 %
Travel bags	1.10 %

The constraints imposed took as a reference the weight on the handbags and are as follows:

$$
\begin{aligned}
0.5 \leq \frac{\mu_{handbags}}{\mu_{bijou}} \leq 2 \quad & 4 \leq \frac{\mu_{handbags}}{\mu_{eveningbags}} \leq 8 \\
2 \leq \frac{\mu_{handbags}}{\mu_{N-textile}} \leq 4 \quad & 4 \leq \frac{\mu_{handbags}}{\mu_{hair}} \leq 8 \\
2 \leq \frac{\mu_{handbags}}{\mu_{textile}} \leq 4 \quad & 4 \leq \frac{\mu_{handbags}}{\mu_{partybags}} \leq 8 \\
2 \leq \frac{\mu_{handbags}}{\mu_{wallets}} \leq 4 \quad & 6 \leq \frac{\mu_{handbags}}{\mu_{travelbags}} \leq 10 \\
2 \leq \frac{\mu_{handbags}}{\mu_{shoes}} \leq 4 &
\end{aligned}
\tag{4}
$$

The first constraint establishes the relationship between the weight attributed to handbag products and bijou, determining that the weight given to handbags can vary between the double of the weight given to bijou products, or it can be at least half that weight. This goes in line with an importance of handbags that is 32 % of total sales, whereas bijou represents 25.5 %. The ratio between these percentages is 1.25. As we wished to allow for some flexibility in the weighting we assumed that this value could be at most 2 and at least 0.5. In the case of the second constraint we impose that the weight assigned to handbags should be at most 4 times and at least 2 times that of the non-textile products. The same reasoning applies to the remaining types of products where a high flexibility was given to the weight of each type of product (but not a complete freedom of choice). In Table 3 we show the average output virtual weights in a DEA model with an without weight restrictions (WR). Note that these are virtual weights and therefore the relationships in (4) cannot be directly inferred.

Table 3 Average weights of outputs with and without weight restrictions

Product	DEA model with WR	DEA model without WR
Textile products	5.70 %	10.70 %
Non-textile products	4.20 %	17.80 %
Hair products	2.90 %	15.40 %
Bijou	58.20 %	13.50 %
Shoes	2.30 %	6.50 %
Handbags	20.70 %	4.10 %
Party bags	0.30 %	5.90 %
Evening bags	0.60 %	8.20 %
Wallets	5.10 %	9.60 %
Travel bags	0.10 %	8.40 %

Table 4 Efficiency statistics with and without weight restrictions

	CRS eff withWR	CRS eff without WR
Average	76.95 %	89.37 %
Standard deviation	16.98 %	13.63 %
No. efficient units	10	28
% Efficient units	15.87 %	44.44 %

When there are no weight constraints the average weight attributed to handbags is 4.10 % and the weight given to travel bags is 8.4 %, which does not reflect the weight of these items in the total sales volume of the average store. When constraints are imposed, the model reflects to a certain extent the importance of each type of product, weighting more bijou items and handbags, while travel items show the lowest weight.

In Table 4 we show the technical efficiency results obtained with and without weight constraints.

The average efficiency obtained without weight constraints is clearly higher, with 28 units appearing 100 % efficient. The weight restricted model shows an average efficiency of about 77 %, and just 10 stores show a 100 % efficiency status. The results with weight constraints are those that will be discussed after this point, as it has been shown that this model provides more discrimination between stores and results in output weights that are more consistent with the importance of each item for the overall sales quantity of the stores.

From the 10 efficient units under the weight restricted model, not all of them are equally important in serving as benchmarks for the inefficient units. The graph in Fig. 2 shows the number of times each efficient unit appears in the reference set of inefficient units.

We conclude from Fig. 2 that stores L53, L22, L36, L52, L26, L56 and L63 are the better performing units, as they show 100 % efficiency and appear 10 or more times in the peer set of inefficient units.

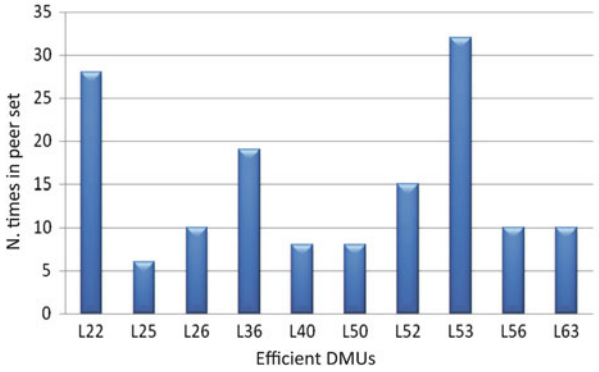


Fig. 2 N. of times efficient units are used as benchmarks of inefficient units

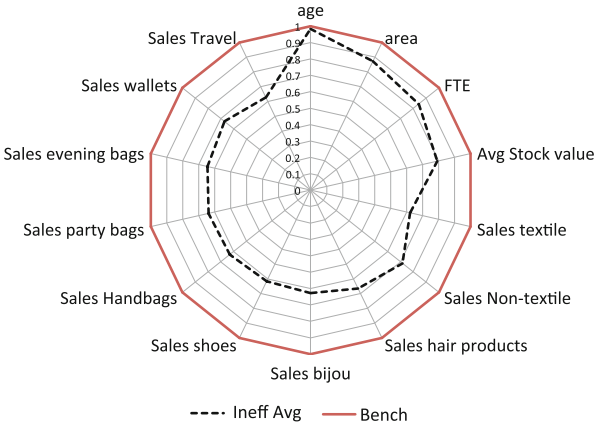


Fig. 3 Inputs and outputs of benchmark and inefficient stores

In Fig. 3 we show the average input and output values observed for the efficient and inefficient stores. Note that values shown in the radars have been normalised by the values of the efficient units (that as a result appear with all variables equal to 1).

From Fig. 3 we can conclude that inefficient stores are in general smaller than efficient stores, but both have a similar age. Inefficient stores have on average a value of FTE, area, and average stock that is about 20 % below the values of efficient stores. With inputs 20 % lower one would expect that outputs produced are also about 20 % lower than those of remaining stores. However, for inefficient stores, outputs are on average lower than those of efficient stores in about 40 %, with the exception of non-textile products where sales quantities of inefficient stores are on average lower than those of efficient stores by 30 %. This is the reason why these stores have been considered inefficient, as they did not produce outputs on the expected proportion for their level of inputs, when compared to other stores in the sample.

Note that these radars can be made for each individual store, to understand their strengths and weaknesses and the factors where they should focus to improve efficiency. Such detailed analysis is not presented in this paper in the sake of brevity.

5.1 The Potential for Improvement of Parfois Stores

The main objective of the efficiency analysis undertaken in this paper is to understand what is the potential for improvement of Parfois stores. We adopt the perspective that this potential should be sought in output sales improvements as inputs are mostly non-discretionary, or difficult for management to act on. In any case the DEA output oriented model allows one to analyse the identified slacks at the input level, which gives us the perspective of inefficiencies that can occur from employing wrong quantities of inputs. Table 5 shows the average improvement potential for outputs for all the stores, calculated as the ratio between the target outputs and actual outputs minus one. The values therefore show how much, on average, inefficient stores can increase their outputs, without changing the volume and mix of inputs employed.

As one can see travel bags show the highest potential for improvement (376.6 %). The value of sales on travel bags for the Parfois stores increased 260 % in 2011, but apparently not all stores followed this trend as when compared to efficient stores, inefficient ones still have a huge potential for improvement. Shoe products are the next items with the largest potential for improvement. This was another category of items showing a big growth in 2011 in Parfois stores in 2011 (49.3 %). The outputs showing the lowest potential for improvement are the non-textile products, the wallets and hair products. This means that inefficient and efficient stores have a closer profile of sales on these products, as it is not on the sales of these products that they show the highest differences. Note however, that a potential for improvement above 30 % is not negligible.

Party bags show a potential for improvement of 72.3 %, and it is known that in 2011 this type of products increased their sales by 39 %, showing that there is still potential for growth on the sales of these products. Bijou is the second most important set of products in Parfois stores, and it still shows potential for improvement in inefficient stores of 52.6 % on average.

The potential for improvement was also analysed in terms of the inputs used, although, as mentioned before, this potential cannot in most cases be realised due to the non-discretionary nature of the inputs. In any case it is of interest to understand what are the input factors that are the sources of largest inefficiencies (Table 6).

The highest potential for input reduction was found in the input age. This is clearly one of the inputs that cannot be changed, but this potential may suggest that some inefficient stores (20 stores) may need some refreshment in terms of their facilities. There is some potential for improvement in the input area too, which seems to suggest that some stores may be over-dimensioned and the company may analyse these stores (17 stores) in detail to see whether some can move to smaller

Table 5 Outputs: potential for improvement

	Textile	Non-textile	Hair	Bijou	Shoes	Handbags	Party bags	Evening bags	Wallets	Travel bags
Average	45.7 %	31.4 %	35.9 %	52.6 %	90 %	56.6 %	72.3 %	49.3 %	34.1 %	376.6 %
StDev	0.315	0.481	0.648	0.537	1.523	0.413	1.093	0.64	0.284	8.206

Table 6 Inputs: potential for improvement

	Area	Age	FTE	Avg Stock
Average	27.42 %	41.27 %	18.81 %	19.47 %
StDev	20.24 %	18.99 %	16.92 %	16.09 %

premises. About 18 inefficient stores appear to be over-staffed as well, as there is some potential for improvement in FTEs. Note, however, that the potential identified here has been the lowest.

5.2 The Effect of Some Factors on Efficiency

After the efficiency analysis we also analysed the relationship between efficiency and other factors that are relevant for the stores. These factors are the number of bijou exhibitors in the store, the concept of the store, the location of the store and the quality of the service, as measured through the visit of a 'mystery client' and her opinion regarding the service of the store.

Parfois stores can have different number of bijou exhibitors (panels and other specific furniture and accessories for displaying the bijou products), which can vary between 4 and 8. It is expected that the higher number of exhibitors affects the sales of bijou and likely the efficiency of the store. Table 7 shows some statistics regarding the number of bijou exhibitors in the store.

From Table 7 we conclude that there appears to be an increasing trend in efficiency with the number of bijou exhibitors in the store, but the number of stores in each group is very different. We performed a Kruskal-Wallis test to evaluate whether the differences in efficiency were statistically significant, and results point for the non-rejection of the null hypothesis of equal distributions of efficiency (p-value of 0.21). However, as far as the sales values of bijou, the hypothesis of equal means is rejected meaning that indeed the sales of bijou tend to increase with the number of exhibitors for bijou in the sore. As a result we conclude that the number of exhibitors is relevant just for the sales of bijou, but the increased sales of bijou for stores with more exhibitors do not necessarily translate in improved efficiency (since efficiency includes the sales of all other articles in the store).

As far as the concept of the store is concerned, Parfois stores can have 5 types of layout. The actual concept is concept V5, which is the one that is used in stores that opened after 2011. This new concept aims at improving the visibility of all products, and creating a cosy and dynamic atmosphere. From the stores analysed only 17 are displayed according to concept V5, 27 in concept V4 and 15 in concept V3. The remaining 4 stores have the oldest layouts (V1 and V2). Given the discrepancy in the number of stores in each group, it was not possible to conclude that the newest concept lead to improved efficiency (through a Kruskal-Wallis test), since stores in concept V4 exhibited on average an efficiency of 79.75 % and stores in concept V5 exhibited an average efficiency of 73.63 %.

Table 7 Bijou exhibitors and efficiency

	4 exhibitors	5 exhibitors	6 exhibitors	7 or more exhibitors
No stores	4	9	40	10
Average eff.	76.00 %	65.68 %	77.77 %	84.22 %
Stantard deviation	16.39 %	21.99 %	15.54 %	15.24 %
% of eff. Units	25 %	0 %	15 %	30 %
Average sales bijou (€000)	22.061	12.769	25.098	30.109
Stantard deviation sales bijou	12.462	6.832	10.553	19.14

Table 8 Location and efficiency

	SC	Street
N	47	16
Average	77.10 %	76.70 %
St Dev	15.50 %	21.40 %
N. efficient stores	6	4
% efficient stores	12.80 %	25 %

As far as the location of the store, these can be located in shopping centers (SC) or in streets. Most of the stores in our sample are shopping center stores (47 out of 63). In Table 8 we show the average efficiency scores for street and shopping center stores. Note that although average efficiency is similar in both groups of stores it is clear that street stores show more variety in efficiency scores, as revealed by a higher standard deviation.

A Wilcoxon test reveals that the hypothesis of equal distributions of efficiency for street stores and shopping center stores is not rejected (p-value of 0.6). Therefore, our sample does not produce evidence that these two types of stores need to be treated separately. Note however that the location of the store has some implications on their area and on the volumes of sales. That is, in general street stores have on average a higher area, but lower FTE personnel and the volume of sales is generally lower.

As far as service quality is concerned, Parfois measures it through 4‘ mystery’ visits realized to each store over the year. We considered as a measure of the service quality the average of the 4 evaluations of the mystery client. The average score of service quality of the Parfois stores is 79 % (in a scale from 0 to 100), where the worst classified store has a score of 49 % and the best classified store shows a score of 93 %.

It is interesting to plot the efficiency of the Parfois stores against the evaluation of the mystery client as shown in Fig. 4.

The correlation coefficient between the two variables is positive and statistically significant but very low (0.273) as it is apparent from Fig. 4. For example, the store with the lowest evaluation by the mystery client is almost efficient (95.47 %).

The objective of this service quality evaluation is to reward store managers that show an evaluation above 70 %. When they show an evaluation lower than that value

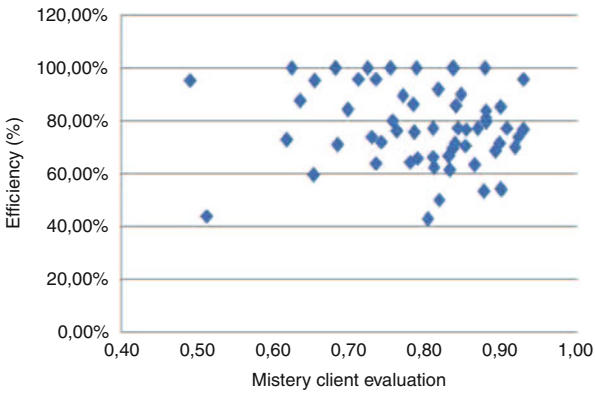


Fig. 4 Efficiency versus quality of Parfois stores

Table 9 Percentage of managers rewarded in 2011

	% Stores	Technical efficiency
Award	81.97 %	77.45 %
Without Award	18.03 %	82.65 %

they are penalized. Table 9 shows the percentage of store managers above and below the 70 % threshold, and the respective average values of efficiency of their stores.

So, we have 18.03 % of shop managers that suffer a penalization on their prizes because they did not achieve the minimum evaluation by the Mystery client. However, it is clear that their stores have an higher efficiency than the stores whose managers were rewarded with a prize for good quality of service. This appears to mean that the company is overemphasizing the way clients are treated within the store and putting the efficiency of the store on the back burner. In addition, when we cross compare the service quality of the store with other variables, like total sales or average sales value, no strong relationships emerge from these comparisons (correlation coefficients are between 0.2 and 0.3). This confirms that indeed, the most efficient stores, the ones that use the least resources to sell the highest quantities of products, are not necessarily the ones that pay more attention to the clients.

6 Conclusion

The performance analysis of Parfois stores is currently done through financial ratios like EBITDA (Earnings Before Interest, Taxes, Depreciation and Amortization) and operational ratios like sales per square meter or per FTE. In addition, Parfois establishes sales objectives for the stores and for the mystery client evaluation. There is also some qualitative evaluation of store managers by their supervisors, which includes an evaluation of their capacity to execute the store merchandizing, their organization capacity, leading teams, punctuality, etc. Every month the stores

are evaluated and rewards are given to their personnel (every member of personnel receives a commission over sales, whose value depends on their position within the store).

The results presented in this paper for the performance of Parfois stores show some discrepancies in relation to the evaluations made by the company in 2011. Although we do not have values for the evaluation performed by the company, we know that managers rewarded show an average efficiency lower than those that were not rewarded, and that the company analyses the performance ratios of stores in absolute terms, not making a comparative analysis between stores.

With the aid of the developed model we conclude that about 85 % of the stores are not efficient, mainly due to low sales in shoes and travel items. Regarding the type of items that traditionally sell more (bijou and handbags), the improvement potential is not as big, but there is room for improvement too. These conclusions should be taken into account by the company, so that it can devise forms to achieve the improvement potential of the stores (e.g. increasing the variety in the number of products displayed, and displaying more references of the products, especially those that show a higher potential for improvement).

The comparison between the efficiency and service quality allowed us to conclude that these are two distinct concepts and the store should eventually reward both separately. The best store will be the one showing a high efficiency and high service quality, but these stores are uncommon. Therefore the company should understand what exactly is being measured by the mystery client and understand whether this evaluation is consistent with the objectives of the stores.

Finally, it is our belief that the company could benefit from the implementation of an evaluation model like the one described in this paper. Several improvements are still possible to the model. For example, the consideration of average prices of the product categories and an analysis of revenue and allocative efficiencies, and also a analysis regarding the location of the store, since eventually the comparative analysis of the stores should be done within the same type of location. As a result, this model is a starting point, for the development of a performance evaluation model at Parfois, that is able to take into account several factors simultaneously and can be used as a means to reward stores in a way that is consistent with the objectives of the company and in a way that can be perceived as fair by the personnel at the stores.

References

1. Athanassopoulos, A., Ballantine, J.: Ratio and frontier analysis for assessing corporate performance: evidence from the grocery industry in the UK. J. Oper. Res. Soci. **46**, 427–440 (1995)
2. Banker, R., Morey, R.: Efficiency analysis for exogenously fixed inputs and outputs. Oper. Res. **34**(4), 51–521 (1986)
3. Banker, D., Lee, S.-Y., Potter, G., Srinivasan, D.: The impact of supervisory monitoring on high-end retail sales productivity. Ann. Oper. Res. **173**, 25–37 (2010)

4. Barros, C.P., Alves, C.: An empirical analysis of productivity growth in a Portuguese retail chain using Malmquist productivity index. J. Retail. Consum. Serv. **11**, 269–278 (2004)
5. Bogetoft, P., Otto, L.: Benchmarking with DEA, SFA, and R. Springer, New York (2011)
6. Charnes, A., Cooper W.W., Rhodes E.: Measuring the efficiency of decision making units. Eur. J. Oper. Res. **8**(2), 429–44 (1978)
7. Färe, R., Grabowski, R., Grosskopf, S., Kraft, S.: Efficiency of a fixed but allocatable input: a non-parametric approach. Econ. Lett. **56**, 187–193 (1996)
8. Grewal, D., Levy, M., Mehrotra, A., Sharma, A.: Planning merchandizing decisions to account for regional and product assortment differences. J. Retail. **75**(3), 405–424 (1999)
9. Ket, H., Chu, S.: Retail productivity and scale economies at the firm level: a DEA approach. Omega **31**, 75–82 (2003)
10. Köksal, C.D., Aksu, A.A.: Efficiency evaluation of A-group travel agencies with data envelopment analysis (DEA): a case study in the Antalya region, Turkey. Tourism Manage. **28**, 830–834 (2007)
11. Korhonen, P., Syrjänen, M.: Resource allocation based on efficiency analysis. Manage. Sci. **50**(8), 1134–1144 (2004)
12. Norman, M., Stoker, B.: Data Envelopment Analysis: The Assessment of Performance. Wiley, Chichester (1991)
13. Thanassoulis, E., Portela, M.C.A.S., Allen, R.: Incorporating value judgments in DEA. In: Cooper, W.W., Seiford, L.M., Zhu, J. (eds.) Handbook on Data Envelopment Analysis. Kluwer Academic Publishers, Boston (2004)
14. Thomas, R., Barr, R., Cron, W., Slocum, J.: A process for evaluating retail store efficiency: a restricted DEA approach. Int. J. Res. Market. **15**(5), 487–503 (1998)
15. Vaz, M.: Desenvolvimento de um sistema de avaliação e melhoria de desempenho no sector do retalho. Dissertação de Doutoramento, Faculdade de Engenharia da Universidade do Porto (2007)
16. Vaz, C., Camanho, A.: Performance comparison of retailing stores using a Malmquist-type index. J. Oper. Res. Soc. **63**, 631–645 (2012)
17. Vaz, C., Camanho, A., Guimarães, R.: Avaliação de desempenho de lojas de retalho utilizando um modelo de Data Envelopment Analysis multi-nivel. Proceedings da IO2009 – 14 congresso da APDIO, pp. 227–234 (2009)

Optimization Clustering Techniques on Register Unemployment Data

Carlos Balsa, Alcina Nunes, and Elisa Barros

Abstract An important strategy for data classification consists in organising data points in clusters. The *k*-means is a traditional optimisation method applied to cluster data points. Using a labour market database, aiming the segmentation of this market taking into account the heterogeneity resulting from different unemployment characteristics observed along the Portuguese geographical space, we suggest the application of an alternative method based on the computation of the dominant eigenvalue of a matrix related with the distance among data points. This approach presents results consistent with the results obtained by the *k*-means.

1 Introduction

Clustering is an important process for data classification that consists in organising a set of data points into groups, called clusters. A cluster is a subset of an original set of data points that are close together in some distance measure. In other words, given a data matrix containing multivariate measurements on a large number of individuals (observations or points), the aim of the cluster analysis is to build up some natural groups (clusters) with homogeneous properties out of heterogeneous large samples [1].

C. Balsa (✉)
Instituto Politécnico de Bragança (IPB), Bragança, and Centro de Estudos de Energia Eólica e Escoamentos Atmosféricos (CEsA) da Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
e-mail: balsa@ipb.pt

A. Nunes
Instituto Politécnico de Bragança (IPB), Bragança, and Grupo de Estudos Monetários e Financeiros (GEMF) da Faculdade de Economia da Universidade de Coimbra, Coimbra, Portugal
e-mail: alcina@ipb.pt

E. Barros
Instituto Politécnico de Bragança (IPB), Bragança, Portugal
e-mail: ebarros@ipb.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_2

Groups are based on similarities. The similarity depends on the distance between data points and a reduced distance indicates that they are more similar. Several distinct methods can be used to measure the distance among the elements of a data set. Along this work we will consider the traditional Euclidian distance, i.e., the 2-norm of the differences between data points vectors.

There are two main classes of clustering techniques: hierarchical and optimization methods. In hierarchical clustering is not necessary to know in advance the number of subsets in which we want to divide the data. The observations are successively included in groups of different dimensions depending on the level of clustering. The result is a set of nested partitions. In each step of the process, two groups are either merged (agglomerative methods) or divided (divisive methods) according to some criteria [2]. In the agglomerative approach, single-members clusters (clusters with only one observation) are increasingly fused until all observations are in only one cluster. The divisive approach starts with a single set containing all points. This group will be increasingly divided as the distance between points is reduced. The set of nested partitions is represented graphically by a dendrogram that has a tree shape indicating the distance's hierarchical dependence.

The *k*-means [3] is an optimization method that partitions the data in exactly *k* clusters, previously determine. This is achieved in a sequence of steps which begins, for instance, with an initial partition randomly generated. In each step the cluster's centroid (arithmetic vector mean) is computed. The minimum distance between each data point and the clusters' different centroids will decide the formation of new clusters. The formation of a new cluster implies assigning each observation to the cluster which presents the lowest distance. After that the centroids are (re)calculated and the former step is repeated until the moment each individual belongs to a stable cluster, i.e., when the sum of the squared distances to the centroid of all data point over all the clusters is minimized. The algorithm presents a rather fast convergence, but one cannot guarantee that the algorithm finds the global minimum [4].

Spectral clustering is also an optimization method. This method is becoming very popular in recent years because it has been included in algorithms used in the identification of the human genome or in web browsers. Beyond biology and information retrieval the method has other fields of application such as image analysis and, in some cases, it can perform better than standard algorithms such as *k*-means and hierarchical clustering [2]. Spectral clustering methods use the *k* dominant eigenvectors of a matrix, called affinity matrix, based on the distance between the observations. The idea is grouping data points in a lower-dimensional space described by those *k* eigenvectors [5]. The approach may not make a lot of sense, at first, since we could apply the *k*-means methodology directly without going through all the matrix calculations and manipulations. However, some analyses show that mapping the points to this *k*-dimensional space can produce tight clusters that can easily be found applying *k*-means [2].

The k-means and spectral methods are rigid because one observation can belong to only one cluster. This rigidity can be avoided by using fuzzy clustering [6]. In this method each observation has a probability of belonging to each cluster, rather than completely belonging to just one cluster as it is the case in the traditional k-means.

Fuzzy k-means specifically tries to deal with the problem where observations are between centroids in an ambiguous position by replacing distance with probability. Thus, one obtain the probability of an observation belonging to each cluster. From the computational point of view this approach is more demanding than traditional k-means. However, it allows more flexibility in the classification of observations.

Most of the observable phenomena in empirical sciences – including the social ones – are of a multivariate nature. It is necessary to deal with huge data sets with high dimensions making sense out of these data and exploring the hidden features of it. In the present research work, spectral clustering is applied in an unusual context concerning the traditional data mining analysis. We classify 278 Portuguese mainland municipalities (*concelhos*) regarding the type/characteristics of unemployment official registers. The set of observations, $x_1, \ldots, x_{278}$, that contains 278 vectors, whose 11 coordinates are the values for some of the indicators used to characterise Portuguese unemployment (gender, age classes, levels of formal education, situation relating unemployment and unemployment duration), is divided in k clusters. The classification of observations resulting from the spectral method is than compared to the classification given by the traditional k-means method.

The results are analysed from both mathematical and economic points of view. The main goal is to find evidence regarding which method produces the best cluster partition and, accordingly, to understand if the resulting clusterisation makes sense in terms of the spatial distribution of unemployment characteristics, over a country's administrative territory. Indeed it is important to understand if the application of the cluster methodology could avoid a priori subjective grouping criteria as the one that just groups municipalities in administrative regions [7].The idea is to understand if a particular cluster methodology for data mining analysis provides useful and suitable information that could be used to the development of national, regional or local unemployment policies. The problem of unemployment has traditionally been studied as a national phenomenon being the national unemployment rates considered as a consequence of national labour market characteristics. However the rates of unemployment at the regional level are very heterogeneous inside countries, particularly in Europe. According to Südekum [8], in Europe, regional labour market disparities within many countries are of about the same magnitude as differences between countries. Taking into account this findings is important to understand the regional dynamics of unemployment [9].

The paper is divided as follows. The k-means method and the spectral clustering method are presented in Sects. 2 and 3, respectively. The methods description is followed by Sect. 4 where data and variables analysed are also presented and described. In Sect. 5 we move ahead toward the optimal number of clusters applying both selected methods. In Sect. 6 the results are presented and discussed, regarding the particular case in which the methodology is applied. Our concluding remarks can be found on Sect. 7.

2 The *k*-Means Method

We are concerned with m data observations $x_i \in \mathbb{R}^n$ that we want classify in k clusters, where k is predetermined. We organize the data as lines in a matrix $X \in \mathbb{R}^{m \times n}$. To describe the k-means method as proposed in [4] we denote a partition of vectors $x_1, \ldots, x_m$ in k clusters as $\prod = \{\pi_1, \ldots, \pi_k\}$ where

$$\pi_j = \{\ell : x_\ell \in \text{cluster } j\}$$

defines the set of vectors in cluster j. The centroid, or the arithmetic mean, of the cluster j is:

$$m_j = \frac{1}{n_j} \sum_{\ell \in \pi_j} x_\ell \tag{1}$$

where n_j is the number of elements in cluster j. The sum of the squared distance, in 2-norm, between the data points and the j cluster's centroid is known as the *coherence*:

$$q_j = \sum_{\ell \in \pi_j} \left\| x_\ell - m_j \right\|_2^2 \tag{2}$$

The closer the vectors are to the centroid, the smaller the value of q_j. The quality of a clustering process can be measured as the *overall coherence*:

$$Q\left(\prod\right) = \sum_{j=1}^{k} q_j \tag{3}$$

The k-means is considered an optimization method because it seeks a partition process that minimizes $Q(\prod)$ and, consequently, finds an optimal coherence. The problem of minimizing the *overall coherence* is NP-hard and, therefore, very difficult to achieve. The basic algorithm for k-means clustering is a two step heuristic procedure. Firstly, each vector is assigned to its closest group. After that, new centroids are computed using the assigned vectors. In the following version of k-means algorithm, proposed by [4], these steps are alternated until the changes in the *overall coherence* are lower than a certain tolerance previously defined.

Since it is an heuristic algorithm there is no guarantee that k-means will converge to the global minimum, and the result may depend on the initial partition $\prod^{(0)}$. To avoid this issue, it is common to run it multiple times, with different starting conditions choosing the solution with the smaller $Q\left(\prod\right)$.

The k-means algorithm

1. Start with an initial partitioning $\prod^{(0)}$ and compute the corresponding centroid vectors $m_j^{(0)}$ for $j = 1, \ldots, k$. Compute $Q(\prod^{(0)})$. Put $t = 1$.
2. For each vector x_i find the closest centroid. If the closest centroid is m_p^{t-1} assign i to $\pi_p^{(t)}$.
3. Compute the centroids $m_j^{(t)}$ for $j = 1, \ldots, k$ of the new partitioning $\prod^{(t)}$.
4. If $\left|Q(\prod^{(t)}) - Q(\prod^{(t-1)})\right| < \text{tol}$, stop; Otherwise $t = t + 1$ and return to step 2.

3 Spectral Clustering Method

Let $x_1, \ldots, x_m$ be a m data observations set in a n-dimensional euclidian space. We want to group these m points in k clusters in order to have better within-cluster affinities and weaker affinities across clusters. The affinity between two observations x_i and x_j is defined by [10] as:

$$A_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right) \tag{4}$$

where σ is a scaling parameter that determines how fast the affinity decreases with the distance between x_i and x_j. The appropriate choice of this parameter is crucial [2]. In [10] we can find a description of a method able to choose the scaling parameter automatically.

The spectral clustering algorithm proposed by [10] is based on the extraction of dominant eigenvalues and their corresponding eigenvectors from the normalized affinity matrix $A \in \mathbb{R}^{m \times m}$. The components A_{ij} of A are given by Eq. 4, if $i \neq j$, and by $A_{ii} = 0$, if $i = j$. The sequence of steps in the spectral clustering algorithm is presented as follows:

The spectral clustering algorithm

1. Form the affinity matrix A as indicated in Eq. 4.
2. Construct the normalized matrix $L = D^{-1/2}AD^{-1/2}$ with $D_{ii} = \sum_{j=1}^{m} A_{ij}$.
3. Construct the matrix $V = [v_1 v_2 \ldots v_k] \in \mathbb{R}^{m \times k}$ by stacking the eigenvectors associated with the k largest eigenvalues of L.
4. Form the matrix Y by normalizing each row in the $m \times k$ matrix V (i.e. $Y_{ij} = V_{ij} / \left(\sum_{j=1}^{k} V_{ij}^2\right)^{1/2}$).
5. Treat each row of Y as a point in $\mathbb{R}^k$ and group them in k clusters by using the k-means method.
6. Assign the original point x_i to cluster j if and only if row i of matrix Y was assigned to cluster j.

4 Data Description

The 278 data observations represents the Portuguese continental *concelhos*. Each data point have 11 coordinates representing characteristics of the unemployed register individuals. Indeed, the unemployed individuals registered in the Portuguese public employment services of the *Instituto de Emprego e Formação Profissional (IEFP)* present a given set of distinctive characteristics related with gender, age, formal education, unemployment spell (unemployment for less than a year or more than a year) and situation related with the unemployment situation (unemployed individual looking for a first employment or for another employment). Due to the methodological particularities of the clustering methods here applied, it should be noted that the characteristics of the individuals registered in each local employment center are not mutually exclusive. If this is the general condition for all variables, it should be stressed that this apply, in particular, to the characteristics of the individuals recorded regarding their labour state within the labour market. A long-term unemployed, for instance, can be looking for a new job or looking for he/she first job. The fundamental feature demanded is the register in a given local labour center for at least 12 months. Of course, is not expected that an individual register presenting an age near the minimum age allowed (18 years) had completed the upper level of formal education but that is not impossible since the upper level of formal education starts counting after the twelve years of study.

The above mentioned characteristics are important determinants of unemployment. For example, the Portuguese labour market is characterised by low intensity transitions between employment and unemployment, and very long unemployment spells [11, 12]. They are also important economic vectors regarding the development of public employment policies. National public policies benefit from being based on simple and objective rules however a blind application of these national policies across space (regions) could be ineffective if the addressed problem is not well explored and identified [13] at a regional level. For example, in many countries the labour market problems of large cities are quite different from those of rural areas – even when the unemployment rate is the same [14]. It is believed this is the case of the Portuguese economy. So well targeted policies are more efficient, in terms of expected results, and avoid the waste of scarce resources. The main strategies of labour market policy have to be varied regionally to correspond to the situation at hand. For instance, it is easier to integrate an unemployed person into a job if the policy measure depends on the local labour market conditions [14].

A complete study of regional similarities (or dissimilarities) in a particular labour market, as the Portuguese, should not be limited by a descriptive analysis of the associated economic phenomena. It should also try to establish spacial comparison patterns among geographic areas in order to develop both national and regional public policies to fight the problem. Indeed high unemployment indicators and regional inequalities are major concerns for European policy-makers since the creation of European Union. However, even if the problem is known the policies dealing with unemployment and regional inequalities have been few and weak [15].

In Portugal, in particular, there are some studies that try to define geographic, economic and social homogeneous groups [16]. Yet, to the best of our knowledge, there are no studies that offer an analysis of regional unemployment profiles. Other economies are starting to develop this kind of statistical analysis using as a policy tool the cluster analysis methodology [7, 17–19].

The data concerning the above mentioned characteristics are openly available in a monthly period base in the website of *IEFP* (http://www.iefp.pt/estatisticas/Paginas/Home.aspx). Additionally, the month of December gives information about the stock of registered unemployed individuals at the end of the respective year. In the case of this research work, data from unemployment registers in 2012 have been used. The eleven variables available to characterise the individuals and that have been used here are divided in demographic variables and variables related with the labour market. These variables are dummy variables, measured in percentage of the total number of register individuals in a given *concelho*, and describe the register unemployed as follows: 1: Female, 2: Long duration unemployed (individual unemployed for more than 1 year), 3: Unemployed looking for a new employment, 4: Age lower than 25 years, 5: Age between 25 and 35 years, 6: Age between 35 and 54 years, 7: Age equal or higher than 55 years, 8: Less than 4 years of formal education (includes individuals with no formal education at all), 9: Between 4 and 6 years of formal education, 10: Between 6 and 12 years of formal education and 11: Higher education.

Women, individuals in a situation of long duration unemployment, younger or older unemployed individuals and the ones with lower formal education are the most fragile groups in the labour market and, consequently, are the most exposed to unemployment situations [20]. They are also the most challenging groups regarding the development of public employment policies, namely the regional ones.

5 Toward the Optimal Number of Clusters

We begin by applying the k-means method to partition in k clusters the data points set $x_1, \ldots, x_m$, with $m = 278$ Portuguese mainland *concelhos* regarding the 11 chosen unemployment characteristics. As the optimal number of targeted groups is unknown a priori, we repeat the partition for $k = 2, 3, 4$ and 5 clusters.

To evaluate the quality of the results from the cluster methodology and to estimate the correct number of groups in our data set we resort the silhouette statistic framework. The silhouette statistic introduced by [1] is a way to estimate the number of groups in a data set. Given observation x_i, the average dissimilarity to all other points in its own cluster is denoted as a_i. For any other cluster c, the average dissimilarity of x_i to all data points in cluster c is represented by $\bar{d}(x_i, c)$. Finally, b_i

denote the minimum of these average dissimilarities $\bar{d}(x_i, c)$. The *silhouette width* for the observation x_i is:

$$s_i = \frac{(b_i - a_i)}{\max\{b_i,\ a_i\}}. \tag{5}$$

The *average silhouette width* is obtained by averaging the s_i over all observations:

$$\bar{s} = \frac{1}{m}\sum_{i=1}^{m} s_i. \tag{6}$$

If the *silhouette width* of an observation is large it tends to be well clustered. Observations with small *silhouette width* values tend to be those that are scattered between clusters. The *silhouette width* s_i in Eq. 5 ranges from -1 to 1. If an observation has a value close to 1, then it is closer to its own cluster than it is to a neighbouring one. If it has a *silhouette width* close to -1, then it is a sign that it is not very well clustered. A *silhouette width* close to zero indicates that the observation could just as well belong to its current cluster or one that is near to it.

The *average silhouette width* (Eq. 6) can be used to estimate the number of clusters in the data set by using the partition with two or more clusters that yield the largest average silhouette width [1]. As a rule of thumb, it is considered that an *average silhouette width* greater than 0.5 indicates a reasonable partition of the data, and a value less than 0.2 would indicate that the data do not exhibit a cluster structure [2].

Figure 1 presents the *silhouette width* corresponding to the case of four different partitions of the data points set, this is, $k = 2, 3, 4$ and 5 clusters resulting from the application of the k-means method.

As it is possible to observe, the worst cases occur, clearly, when $k = 3$ and $k = 5$. For these cases, some clusters present negative values and others appear with small (even if positive) silhouette indexes. In the case of $k = 2$ and $k = 4$ clusters there are no negative values, however we find large silhouette values mostly in the case of the two clusters partition.

To get a single number that is able to summary and describe each clustering process, we find the *average of the silhouette* values (Eq. 6) corresponding to $k = 2,\ 3,\ 4$ and 5. The results can be observed in Fig. 2.

The two cluster solution presents an average silhouette value near 0.44 and the four cluster solution presents an average silhouette value near 0.29. These results confirm the ones above. The best partition obtained with the application of the k-means method occurs with $k = 2$. Nonetheless, the *average of the silhouette* is close but smaller than 0.5 which reveals that the data set does not seem to present a strong trend to be partitioned in two clusters.

Figure 3 shows the *silhouette width* corresponding to each observation in the case of four different partitions of the data set points. This is, in $k = 2, 3, 4$ and 5 clusters, resulting from the application of the spectral clustering method.

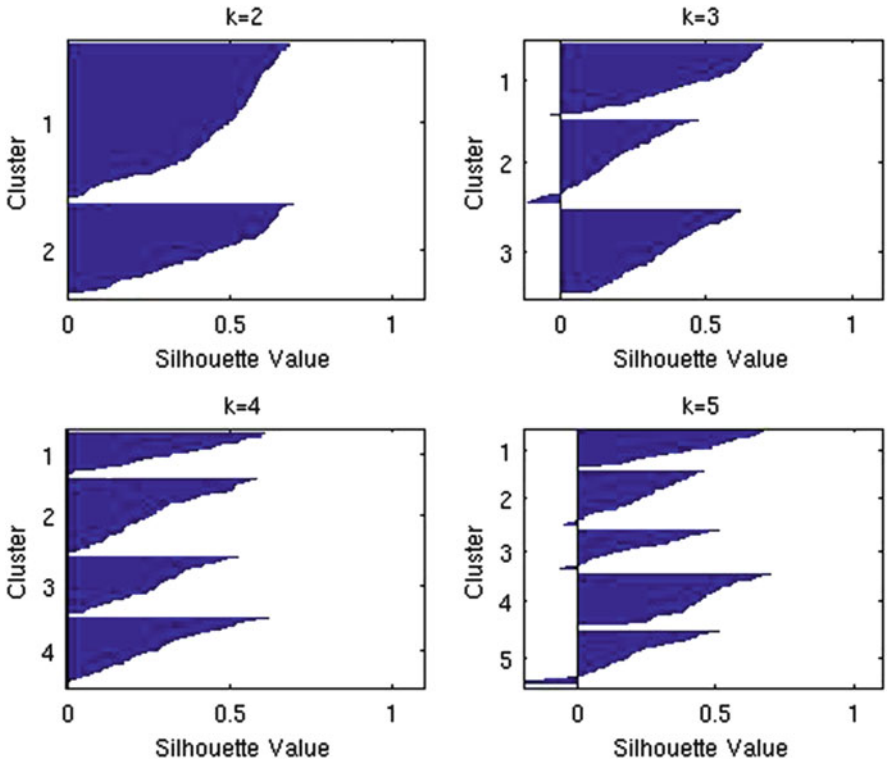


Fig. 1 *Silhouette width* for $k = 2, 3, 4$ and 5 clusters resulting from the *k*-means method

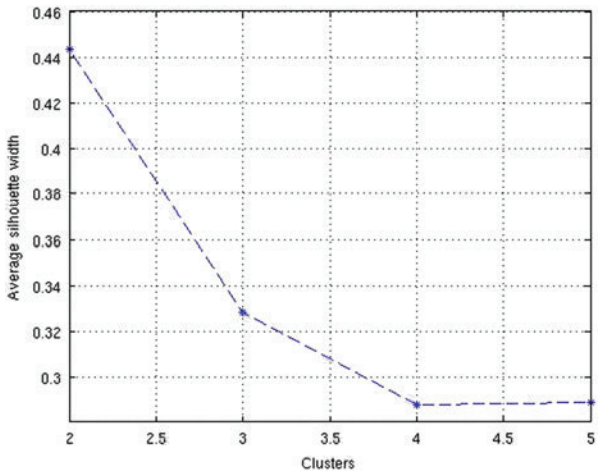


Fig. 2 *Average silhouette width* for $k = 2, 3, 4$ and 5 clusters resulting from the *k*-means method

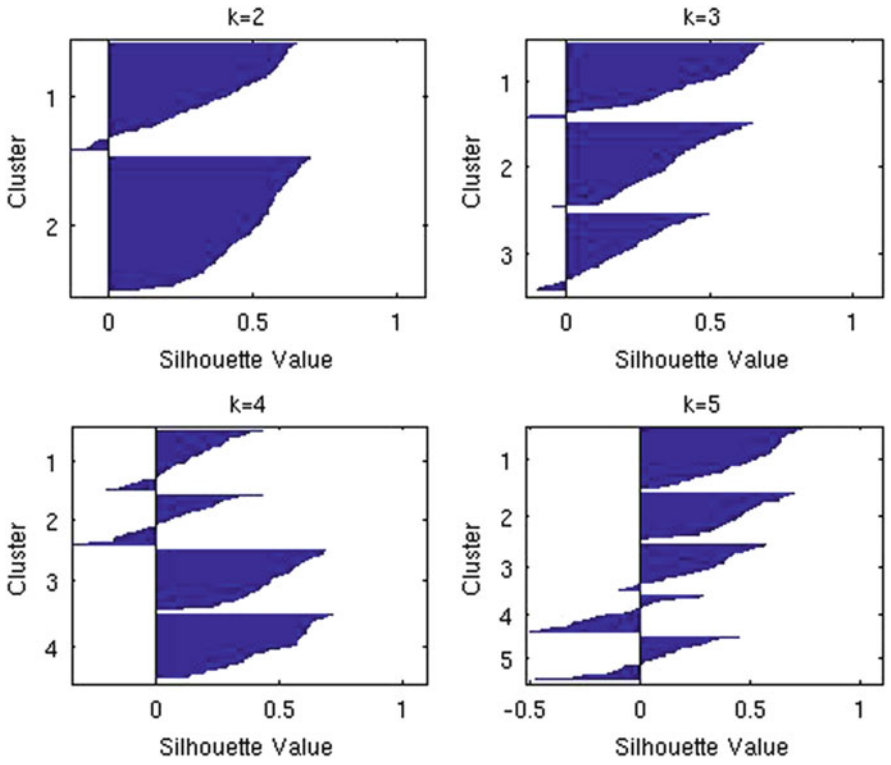


Fig. 3 *Silhouette width* for $k = 2,\ 3,\ 4$ and 5 clusters resulting from the spectral method

In this case all the tested partitions present clusters where can be observed negative values. The worst cases occur, clearly, when $k = 4$ and $k = 5$. Here we get values close to -0.5. In the case $k = 3$ is possible to observe negative values in the three cluster obtained whereas in the case of $k = 2$ the negative values are just observed in one of the two clusters.

The trend observed with the *silhouette width* is confirmed by the *average of the silhouette* values corresponding to the spectral clustering process with $k = 2,\ 3,\ 4$ and 5 clusters (Fig. 4).

The two cluster solution has an average silhouette value near 0.43 and decrease as the number of clusters increases. The best partition obtained with the spectral clustering method occurs with $k = 2$. These results are in agreement with the partitioning found by using the *k*-means method. The *average of the silhouette* value (0.43) is very close to the one calculated with *k*-means method (0.44).

As mentioned before, the results obtained with the *k*-means method agree with the results obtained with the application of the spectral methods. The best partition of the data set is accomplished with two clusters. However, this trend is not completely crystal clear. Indeed, the *average of the silhouette* in the two cases is

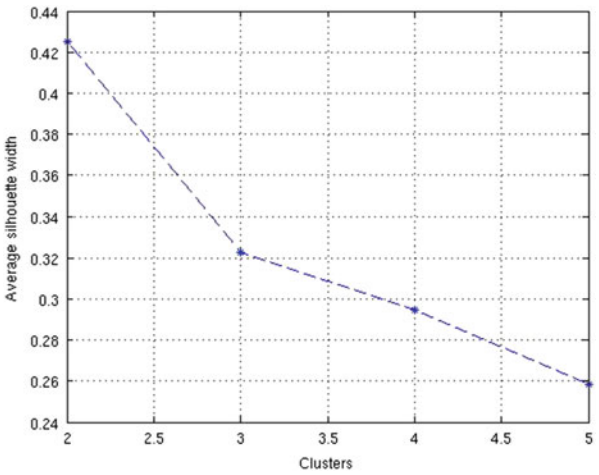


Fig. 4 *Average silhouette width* for $k = 2, 3, 4$ and 5 clusters resulting from the spectral method.

smaller than 0.5. The computed value indicates that the distance between the two considered clusters is not very large.

6 Mathematical and Economic Results' Analysis

Both spectral clustering method and k-means method indicate that the data are best partitioned into two clusters. The statistical properties of theses two clusters are presented in Table 1.

Despite the number of observations in each cluster is not the same, it appears that for both methods the first cluster is the largest. This is, includes a bigger number of *concelhos*: $n_1 = 177$ for the k-means and $n_1 = 154$ for the spectral method. The difference of 23 observations for the first cluster is reflected in the computed local coherence q (Eq. 2) that is larger for the k-means methods ($q_1 = 3.4161$). The second cluster comprises $n_2 = 101$ observations and presents a local coherence of $q_2 = 1.9115$, for the k-means, and $n_2 = 124$ observations and a local coherence of $q_2 = 2.6026$ for the spectral method. Although the differences between the computed coherence for each cluster, we can observe that both methods achieve a very similar overall coherence (Eq. 3), $Q \approx 5.3$ for the k-means and $Q \approx 5.4$ for the spectral method. The results presented in Table 2 show that clusters obtained by the two methods are very similar. We can observe that 153 observations are assigned to the first cluster and 118 assigned to the second by the two methods. There are only 7 observations whose allocation fluctuates with the method. This number represents about 2.5 % of the total number of observations (278). This means that the uncertainty associated with the formation of the two clusters is small.

Table 1 Statistical properties of the two clusters resulting from *k*-means and spectral methods

Method	j	n_j	q_j	Q
k-means	1	177	3.4161	5.3276
	2	101	1.9115	
Spectral	1	154	2.7511	5.3536
	2	124	2.6026	

Table 2 Repeated observations in each cluster

Cluster j	*k*-means n_j	Spectral n_j	Repeated n_j
1	160	153	153
2	118	125	118

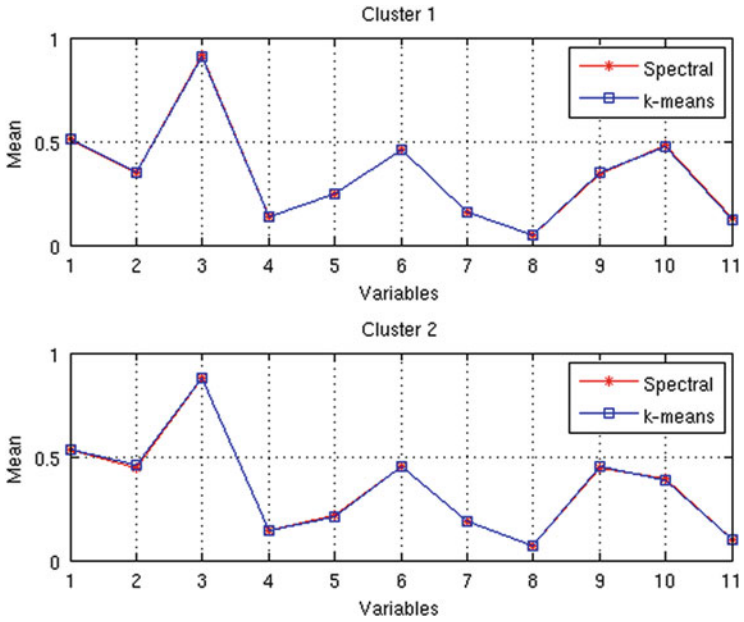


Fig. 5 Mean values computed for the two methods by cluster

For a more complete comparison analysis of the results obtained by *k*-means and spectral methods, it is also important to analyse two distribution measures: mean and standard deviation. The measures are presented for each one of the 11 variables used in the cluster analysis. In Fig. 5 we compare the mean value obtained for the 11 parameters that characterise the two clusters obtained by the two clusterisation methods. In Fig. 6 we compare the standard deviation value. Note that in these two figures the comparison analysis is done regarding the cluster methods applied.

It is visible that the computed mean values, regarding each one of the variables, are very similar in the two clusters independently of the cluster method used. This situation is not unusual in times of economic crisis. In these periods of the economic

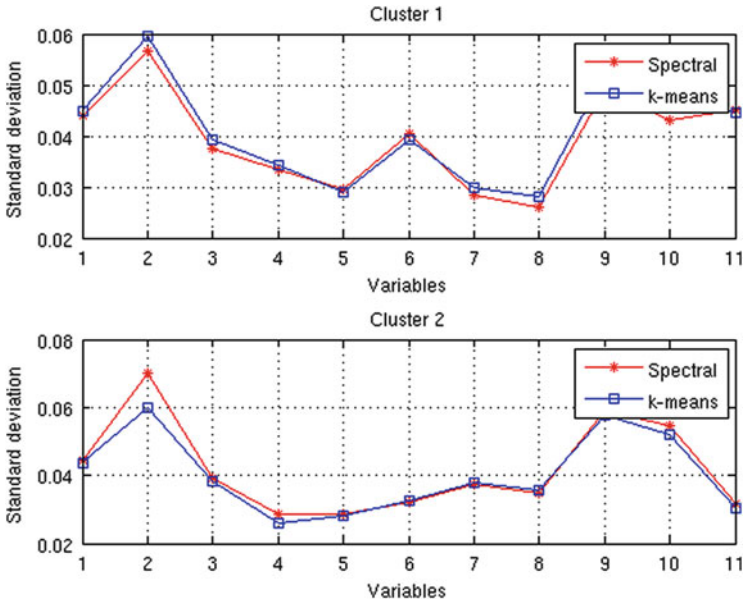


Fig. 6 Standard deviation values computed for the two methods by cluster

cycle the registered unemployment increases, in general, not sparing any particular group. So, the average values of registers, by characteristic, tend to converge. For the computed standard deviation values we can observe a first cluster where the standard deviation, for the overall set of variables, are slightly higher for the *k*-means and a second cluster where the observed trend is reversed. In short, we can observe that the results for both methods are similar regarding the measure of central tendency of each one of the variables but the variability of values, regarding the central tendency, differ between cluster methods.

The mean and standard deviation measures can be compared regarding the values computed by cluster. From this point of view the analysis would have an economic focus. So, in Fig. 7 we compare the mean value obtained for the 11 parameters for each one of the clusters by cluster method. In Fig. 8 we compare the computed standard deviation value. The lecture of both figures should not forget the observation made on the data description – a register in a variable do not excludes the register in an other variable since they are not mutually exclusive.

From the Figs. 7 and 8 it is possible to observe that both methods retrieve clusters that present the same pattern. In the second cluster (cluster 2) are gathered the Portuguese mainland *concelhos* that present a higher percentage of unemployed register individuals with more problematic characteristics – women, long duration unemployed individuals, individuals that are looking for a job for the first time (individuals with no connections with the labour market), individuals with more than 55 years and with lower number of years of formal education (for example, this

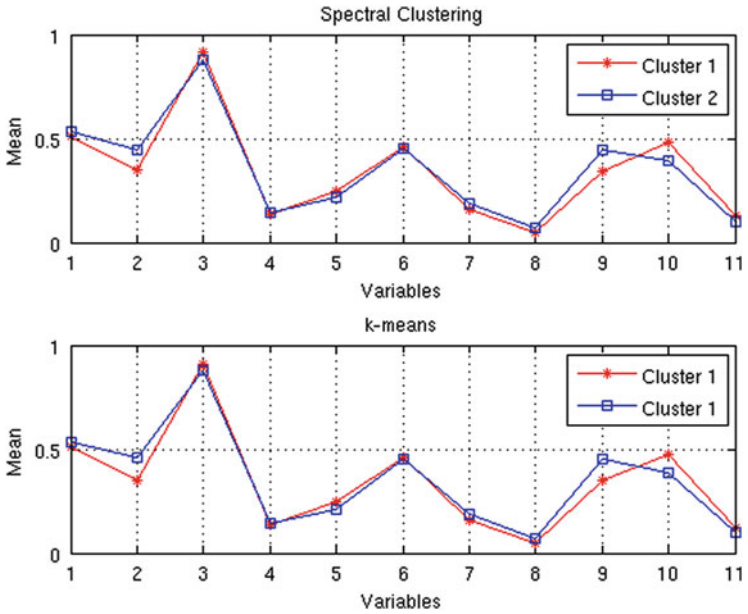


Fig. 7 Mean values computed for the two clusters resulting from *k*-means and spectral method

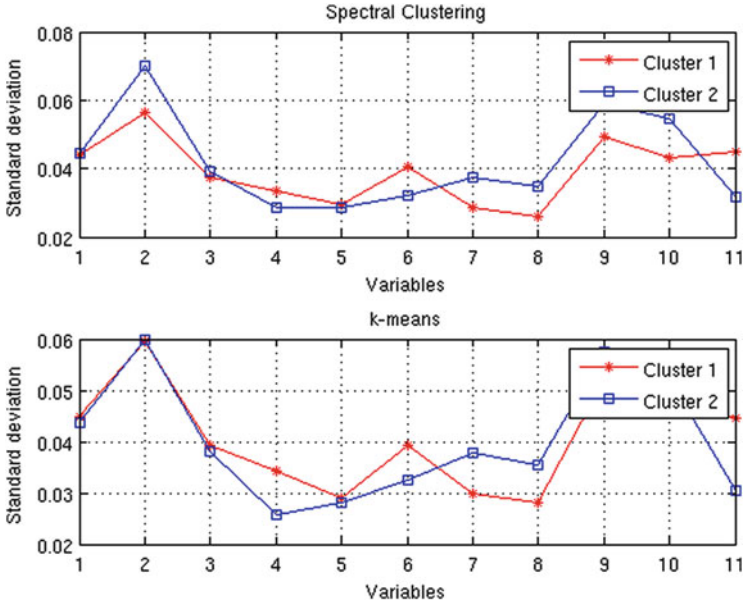


Fig. 8 Standard deviation values computed for the two clusters resulting from *k*-means and spectral method

cluster gathers the *concelhos* with a lower percentage of unemployed individuals with a higher education). As mentioned before these groups of individual are the most fragile labour market groups. Both cluster methods seem to divide the total number of *concelhos* in two economic meaningful clusters. Despite the stage of the economic cycle, that tends to align the unemployment registration rates, regardless of the observed individual characteristics, is possible to verify the existence of regional differences that should be studied and analysed carefully in order to make employment public policies more effective and efficient. The success of labour policies depends on the regional labour market conditions. As a consequence, first, policy-makers should be very careful in promoting those policies since their effectiveness might significantly vary. Second, policy-makers should adjust labour policy strategy to the regional economic structure. It follows that when designing a labour market strategy, the economic context should be heavily taken into account [21].

Regarding the standard deviation we observe that the *k*-means method retrieve clusters that present a lower variability among the observations in each cluster, by variable. The variability seems to be lower for the overall set of characteristics even if the *k*-means method divides the total number of observations in more uneven clusters.

7 Concluding Remarks

In short, both methods denote the same data partition. Applying both methods, the data partition into two clusters minimises the dispersion of data values. The use of the spectral clustering method in an unusual economic application shows potential benefits. Without algorithm parameters refinement the method presented results that are consistent with the *k*-means results.

From the economic point of view both methods show the importance of dividing Portuguese *concelhos* in two well defined spatial groups which could be object of distinct public policies and of particular unemployment measures. Well targeted labour market measures are, recognisable, more efficient with the cluster methodology helping the identification of different and well defined target regions – regions with similar characteristics and problems. Indeed the allocation of unemployment particular measures according to a multivariate classification as the one explored in this paper brings benefits that should not be ignored. The classification obtained (the classification enables employment offices to compare themselves with others in the appropriate peer group)can be used, for instance, to assess and support the labour market policy adopted by each region. Although differences remain regarding labour market conditions the complexity of reality is reduced – is possible to differentiate within types of registered unemployed individuals since the results for the distance matrix between all labour market regions is available [14],

As pointed by Campo and co-authors [13] in their work, it is important to conduct further analysis aiming to compare results from different techniques, data regarding different moments of the economic cycle and different unemployment variables.

References

1. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
2. Martinez, W.L., Martinez, A.R., Solka J.L.: Exploratory Data Analysis with MATLAB. CRC, Boca Raton (2010)
3. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press (1967)
4. Eldén, L.: Matrix Methods in Data Mining and Pattern Recognition. SIAM, Philadelphia (2007)
5. Mouysset, S., Noailles, J., Ruiz, D.: Using a global parameter for Gaussian Affinity matrices in spectral clustering. In: Palma, J,M.L.M., Amestoy, P.R., Daydé, M., Mattoso, M., Lopes, J.C. (eds.) High Performance Computing for Computational Science – VECPAR 2008. Lecture Notes in Computer Science, vol. 5336, pp. 378–390. Springer, Berlin/Heidelberg (2008). ISBN: 978-3-540-92858-4, doi:10.1007/978-3-540-92859-1_34, http://dx.doi.org/10.1007/978-3-540-92859-1_34
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
7. Álvarez de Toledo, P., Núñez, F., Usabiaga, C.: Labour market segmentation, clusters, mobility and unemployment duration with individual microdata. MPRA Paper 46003, University Library of Munich (2013)
8. Südekum, J.: Increasing returns and spatial unemployment disparities. Papers Reg. Sci. **84**, 159–181 (2005)
9. Garcilazo, J.E., Spiezia, V.: Regional unemployment clusters: neighborhood and state effects in Europe and North America. Rev. Reg. Stud. **37**(3), 282–302 (2007)
10. Ng, A.Y., Jordan, M.I., Weiss Y.: On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. (NIPS) **14**, 849–856 (2002)
11. Carneiro, A., Portugal, P., Varejão, J.: Catastrophic job destruction during the Portuguese economic crisis. J. Macroecon. **39**, 444–457 (2014)
12. Blanchard, O., Portugal, P.: What hides behind an unemployment rate: comparing Portuguese and U.S. labor markets. Am. Econ. Rev. **91**, 187–207 (2001)
13. Campo, D., Monteiro, C.M.F., Soares, J.O.: The European regional policy and the socio-economic diversity of European regions: a multivariate analysis. Eur. J. Oper. Res. **187**, 600–612 (2008)
14. Blien, U., Hirschenauer, F., Van, P.t.H.: Classification of regional labour markets for purpose of labour market policies. Pap. Reg. Sci. **89**(4), 859–881 (2009)
15. Overman, H.G., Puga, D.: Unemployment clusters across Europe's regions and countries. Econ. Policy **17**(34), 115–148 (2002)
16. Soares, J.O., Marques, M.M.L., Monteiro, C.M.F.: A multivariate methodology to uncover regional disparities: a contribution to improve European union and governmental decisions. Eur. J. Oper. Res. **45**, 121–135 (2003)
17. Arandarenko, M., Juvicic, M.: Regional labour market differences in Serbia: assessment and policy recommendations. Eur. J. Comp. Econ. **4**(2), 299–317 (2007)
18. López-Bazo, E., Del Barrio, T., Artís, M.: Geographical distribution of unemployment in Spain. Reg. Stud. **39**(3), 305–318 (2005)

19. Nadiya, D.: Econometric and cluster analysis of potential and regional features of the labor market of Poland. Ekonomia 21:28–44 (2008)
20. Dean, A.: Tackling Long-Term Unemployment Amongst Vulnerable Groups. OECD Local Economic and Employment Development (LEED) Working Paper 2013/11. OECD Publishing (2013)
21. Altavilla, C., Caroleo, F.E.: Asymmetric effects of national-based active labour market policies. Reg. Stud. **47**(9), 1482–1506 (2013)

Web Based Application for Home Care Visits' Optimization of Health Professionals' Teams of Health Centers

Bruno Bastos, Tiago Heleno, António Trigo, and Pedro Martins

Abstract Health Centers have among one of their many tasks the provision of health care at home. This service is provided by teams of health professionals, usually composed of physicians and nurses belonging to the Health Centers. The scheduling of the visits is made by a health professional that groups one or more routes in order to minimize the time of team's visits. However, as there is no technical or computer application to plan and optimize the visits in a systematic way, the obtained solutions are rarely the best ones. To improve this situation we were challenged by a Health Center to create an application to optimize the visits of health care professionals. This paper presents the solution developed involving a web application called "Health at Home", which uses the heuristic of Clarke and Wright. The main novelty of this work is the inclusion of priority and non-priority patients, according to their degree of aseptic, within the routes optimization of the health professionals' teams.

B. Bastos (✉) • T. Heleno
Instituto Politécnico de Coimbra, ISCAC, Quinta Agrícola, Bencanta, 3040-316 Coimbra, Portugal
e-mail: iscac10184@alumni.iscac.pt; iscac10178@alumni.iscac.pt

A. Trigo
Instituto Politécnico de Coimbra, ISCAC, Quinta Agrícola, Bencanta, 3040-316 Coimbra, Portugal

Centro ALGORITMI, Universidade do Minho, Guimarães, Portugal
e-mail: aribeiro@iscac.pt

P. Martins
Instituto Politécnico de Coimbra, ISCAC, Quinta Agrícola, Bencanta, 3040-316 Coimbra, Portugal

Centro de Investigação Operacional, Universidade de Lisboa, Lisboa, Portugal
e-mail: pmartins@iscac.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_3

1 Introduction

The provision of home health care is becoming one of the most important and increasing areas in Europe due to the population's aging and to the fact that it is economically advantageous to have people at home instead of having them in a hospital bed [10].

The home-based care provided by public or private entities has been the subject of recent research mainly in the operations research area with particular attention on route's optimization and on the staff team's composition that provide that kind of services [2, 3, 10, 12].

In providing home-based care, Health Centers (HC) play a very important role since they are closer to population than hospitals, which are more focused on serving patients that visiting them. To play this role, Health Centers have to schedule the professional health teams (doctors and/or nurses) and the routes of those teams to visit patients in their homes.

With the aim of optimizing these routes, it was proposed us by a HC's nurse from an urban area (Coimbra, Portugal) the development of an application being able to respond adequately to this challenge. In this HC in particular the teams that do the visits are the ones that plan the routes using their geographical knowledge about the area in which they operate, thus defining the route they consider to be the best, which is not always true. The odds of the route obtained not being the best one increases when one is forced to attend certain restrictions, like, for example, the issue about the priority or aseptic patients.

The Vehicle Routing Problem (VRP) is a well-known combinatorial optimization problem. Its most common objective is to find the shortest path or route to, for example, the delivering or picking of goods [1] or persons [11]. In the particular case of the professional health teams, they do not deliver goods but medical procedures to sick patients, like the administration of injectable. It is assumed that the times of travel between the patients homes' locations and between these locations and the HC are known, that the times of the visits are also known and that the professional health teams always return to the HC (the origin).

Figure 1 shows the possible locations of the visits (patient's houses) done during a day by the professional health teams of a Coimbra's HC. The red dots show the locations that teams have to visit. The point with a red circle around is the CS from where the teams start the visits and must obligatorily return. In section five, devoted to the development of the application, a small example is presented in order to show the use of the application.

This paper is structured as follows: the next section describes the home-based care environment; the third section presents the mathematical formulation for the problem; the fourth section presents the methodology used to solve the problem, based on Clarke and Wright's heuristic and a variation using the second order heuristic; the fifth section presents the developed web application – Health at Home – using the PHP language and MySQL database by Oracle; and finally the last section presents some concluding remarks and proposals for future work.

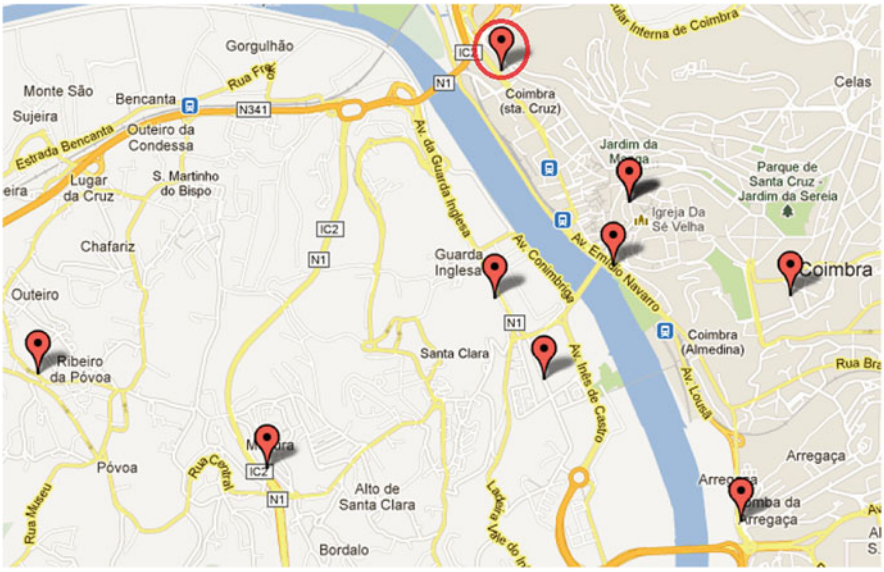


Fig. 1 Home visits locations in the city of Coimbra

2 Home-Based Care

Home-based care is a type of care provided continuously, pointing to the resolution of some health problems, whose complexity does not require hospitalization, but given the global dependence situation, transient or chronic, prevents patients to travel to the HC [9]. In [8] it is possible to see some of the reasons for requesting this kind of health care, which includes, among others, chronic obstructive pulmonary diseases and congestive heart failure.

The procedures/treatments delivered at home are many [8], including for example, at nursing care level, the Inhalation Therapy, Suctioning, Fluid Therapy, Wound Control/Pressure Ulcers, Noninvasive ventilation (NIV), Oxygen Therapy, Enteral Feeding, etc.

Although people's houses do not present risk of contamination by microorganisms existing in hospitals [9], health professionals are aware that it is mandatory to prevent such infections, which can be harmful to weaker people. In this regard and in order to reduce contamination by microorganisms, the present work takes into account the asepsis degree in the definition of the patient's type.

2.1 Patient's Type (or Asepsis Degree)

Asepsis is the set of measures that are taken to prevent the contamination by microorganisms in an environment that does not have them, which means that an aseptic environment is one that is free from infection.

Antisepsis is the set of proposed measures to inhibit the growth of microorganisms or remove them from a certain environment, which may or may not destroy them, by using antiseptics or disinfectants.

Aseptic (or priority) patients are those with a more weakened immune system and therefore need to be the first to be visited in an attempt to minimize potential contamination. So, in the designing of health professionals' visits route, such patients can never come after non-priority patients to avoid to the risk of contamination by microorganisms.

3 Mathematical Formulation

The problem being handled in the present paper can be seen as a variant of the VRP, in which the routes are bounded by time constraints and the set of nodes is partitioned in two subsets: priority and non-priority nodes. The prioritized nodes are associated to patients with high asepsis degree and the non-prioritized nodes include patients who do not require special aseptic precautions.

For this problem, we consider the directed graph $G = (V \cup \{0\}, A)$, with V the set of nodes (patients), 0 the origin node (HC) and A the set of arcs (i, j) between every pair of nodes $i, j \in V \cup \{0\}$. Each arc (i, j) has an associated parameter t_{ij} which describes the travel time from i to j. We also denote by parameter r_i the time for visiting client i, for all nodes $i \in V$.

Each route starts and ends at the origin, node 0, and the route is defined by a sequence of arcs. The total time of the route is the sum of the times of all its arcs (t_{ij}) and the sum of the times of its nodes (r_i). There is a time limit for the total time of a route, being denoted by T.

In this problem, we are intending to calculate a set of routes that ensure the visit to all nodes (patients), respecting the aseptic degree restriction and the total limit time for each route, with the aim of minimizing the sum of the total time of all the routes involved.

When looking for the times to travel between locations, and attending to the practical problem in hand, we could treat them in a symmetric sense. However, and for the purpose of formulating the problem, we are going to use an asymmetric version of the original graph, where each edge $\{i, j\}$ is substituted by the pairs (i, j) and (j, i). There are two reasons for undertaking this option: (i) because we can obtain a stronger flow based (compact) formulation from the linear programming relaxation stand point; and (ii) because the chance to distinguish between traveling

from i to j or from j to i allows us to model the priorities among the nodes in a simpler way, using known oriented and compact formulations for the VRP.

In fact, the problem can be adapted from a capacitated VRP, simply by removing the variables associated to the arcs (i,j) with $i \in V^2$ and $j \in V^1$, for V^1 and V^2 a partition of the set of nodes (clients) V, where V^1 and V^2 represent the priority and the non-priority clients, respectively.

While in the classic capacitated VRP the flow variables reflect the goods accumulated/released in the graph's nodes, in this version the flow variables represent the times accumulated in the arcs and nodes, along the route.

Therefore, the following formulation is propose:

- Variables:
  - design variables:

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i,j) \text{ belongs to the solution} \\ 0, & \text{otherwise} \end{cases}, \text{ for } i,j \in V \text{ and } i \neq j$$

  - flow variables:
    $y_{ij} \equiv$ accumulated time flow after traveling through arc (i,j) and before starting the service at j, for $i,j \in V$ and $i \neq j$
- Parameters:
  - $t_{ij} \equiv$ travel time from i to j, for $i,j \in V$ and $i \neq j$
  - $r_i \equiv$ visit time in client i, for $i \in V$ (assuming that $r_0 = 0$)
  - $V = V^1 \cup V^2$ and $V^1 \cap V^2 = \emptyset$, where V^1 is the set of nodes associated to priority clients and V^2 the set of nodes associated to non-priority clients
  - T is the maximum total time for each route
  - Node 0 is the origin, representing departure/arrival point for all routes
- Objective function:

$$min \;\; z = \sum_{i \in V} y_{i0} \tag{1}$$

- subject to the following constraints,

$$\sum_{i \in V \cup \{0\}} x_{ij} = 1, \; j \in V \tag{2}$$

$$\sum_{j \in V \cup \{0\}} x_{ij} = 1, \; i \in V \tag{3}$$

$$\sum_{i \in V \cup \{0\}} y_{ji} - \sum_{i \in V \cup \{0\}} y_{ij} - \sum_{i \in V \cup \{0\}} t_{ji} x_{ji} = r_j, \; j \in V \tag{4}$$

$$(t_{ij} + r_i)x_{ij} \leq y_{ij} \leq (T - r_j)x_{ij},\ i,j \in V \cup \{0\} \tag{5}$$

$$x_{ij} \in \{0,1\},\ y_{ij} \geq 0,\ i,j \in V \cup \{0\} \tag{6}$$

For simplicity of notation, we have not isolated in the model all variables x_{ij} and y_{ij} for $i \in V^2$ and $j \in V^1$ and all variables x_{ii} and y_{ii} for $i \in V \cup \{0\}$. In effect, the correct formulation for the problem in hands does not include those variables.

Constraints (2) and (3) guarantee that to each node (client) arrives a team (route) and a single one and that exactly one team (route) leaves from each node (client). The flow conservation constraints (4) guarantee the temporal increment of each route. When embedded in the entire model, these equalities can avoid the creation of sub-circuits among the nodes in V. Inequalities (5) establish the link between the variables x and y and guarantee that the maximum cumulative flow on any route does not exceed the time limit T.

Note that the fact of having flow variables (and design variables, as well) linking nodes from set V^1 to V^2 and having no variables in the opposite direction, guarantees that it will always travel from the priority to the non-priority clients and never the reverse, when traveling between clients from one set to the other.

The objective function minimizes the total time length of all routes in the solution characterized by the final flow variables in the routes.

3.1 Using the Solver to Obtain the Solution

The formulation described in Sect. 3 was solved using the ILOG/CPLEX 11.2 package, and all experiments were performed under Microsoft Windows 7 operating system on an Intel Core i7-2600 with 3.40 GHz and 8 GB RAM.

The mixed integer programming (MIP) algorithm of CPLEX was used for solving the mathematical models. Most default settings were considered, which involve an automatic procedure that uses the best rule for variable selection and the best-bound search strategy for node selection in the branch-and-bound tree.

A number of computational tests were performed using the 5 nodes instance proposed in the forthcoming Sect. 4.2. The example furthered assumes that $r_i = 1$ for all $i \in V$ and that $V^1 = \{1,3\}$ and $V^2 = \{2,4,5\}$, thus 1 min is spent in each client. Nodes 1 and 3 represent priority clients. With these parameters, the model was solved considering three different values for the time limit parameter T, setting $T = 10, 12$ and 15. The times to reach the optimums were lower than 0.01 s in all the three attempts. The optimum solutions ($z*$ represents the optimum solution value) are presented in Table 1.

As expected, the number of routes increase for lower values of the routes' time limit, leading to larger optimum solution values, that is, the entire solution becomes more time demanding. In addition, the solutions respect client's priority hierarchies, putting both clients 1 and 3 to be visited before all those in $\{2,4,5\}$.

Table 1 Solver solution of example in Table 2

T	$z*$	Solution
10	24	$0 \rightarrow 2 \rightarrow 0$
		$0 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 0$
		$0 \rightarrow 5 \rightarrow 0$
12	19	$0 \rightarrow 2 \rightarrow 4 \rightarrow 0$
		$0 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 0$
15	15	$0 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 0$

All these instances were solved very fast to optimality. However, when the size of the problem grows, namely when the number of clients increase, we may loose the chance to reach the exact solution within reasonable execution time. This is something observed in many works in the literature that also resort to flow based (compact) models for solving the VRP. Even if using Miller-Tucker-Zemlin subtour elimination constraints instead of flow conservation ones (see, e.g., [5]), the conclusion would be similar when attempting to solve large sized problems. The reason is common to many other problems. In effect, this is also an NP-hard problem, which means, in a broad sense, that, so far, there is no efficient method for solving the problem. For this reason we propose in the forthcoming sections approximate methods for addressing the problem, resorting to Clark and Wright based heuristics.

4 Methodology

As a first approach to implement an algorithm to solve this problem the Clarke and Wright's heuristic [4] was chosen. In a second phase, it was decided to improve the algorithm, first with the introduction of some random feature and then using the Clarke and Wright's heuristic in a repetitive method, namely using a second order algorithm [6, 7]. The combination of heuristics is a common practice for handling hard combinatorial optimization problems [3].

4.1 Clarke and Wright's Heuristic

This heuristic, also known as the savings algorithm, was first proposed in 1964 by Clarke and Wright for solving capacitated vehicle routing problems, being denoted as the Capacitated Vehicle Routing Problem (CVRP). This initial form of the VRP involves an unlimited and homogeneous fleet of vehicles [4, 5]. Next, it is described the Clarke and Wright heuristic adapted to the problem under discussion.

- Clarke and Wright [5] adapted to the problem:
  1. Calculate the savings

$$s_{ij} = t_{i0} + t_{0j} - t_{ij}, \text{ for } i, j = 1, \ldots, n \text{ and } i \neq j \tag{7}$$

     – with the exception of savings s_{ij} such as $i \in V^2$ and $j \in V^1$;
  2. Sort savings in descending order;
  3. Go trough the savings' list starting at the first element. For each element arc (i, j) with $s_{ij} > 0$ and having connections $(i, 0)$ and $(0, j)$, temporarily merge the two routes by linking node i to node j. Check the viability of the route obtained and if so remove arcs $(i, 0)$ and $(0, j)$. Go to the very next putaitve connection (i, j) in the savings' list until no further improvement is possible.

To summarize, the algorithm starts by making a list of all possible round-trip routes from an origin (HC) to each of the destinations (patients), then it calculates the savings using Eq. (7). At each step, two routes are merged according to the greater savings that can be generated and the viability of the new connection is checked. The algorithm stops when there are no more feasible and positive savings in the list, showing that it cannot improve the current solution any further.

4.2 Example

Consider the example with a HC (*HC*) and five patients P_1, P_2, P_3, P_4 and P_5 two of them priority ones (the P_1 and P_3) and the remaining non-priority.

Table 2 shows the times between different nodes and Table 3 presents the savings calculated using the Eq. (7), with the exception of savings s_{ij} such that $i \in V^2$ and $j \in V^1$. Once again, it was assumed that $r_i = 1$ for all $i \in V$ and that $V^1 = \{1, 3\}$ and $V^2 = \{2, 4, 5\}$, thus, the teams spend 1 min in each client.

Note that although the matrix distance is symmetric, our algorithm is asymmetric, as it takes into account the directions of flows, so that at the beginning all savings are taken into account.

Table 2 Time (minutes)

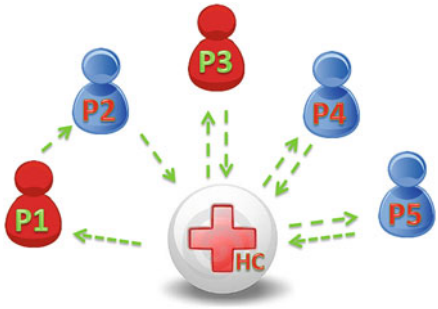
t_{ij}	*HC*	P_1	P_2	P_3	P_4	P_5
HC		5	4	2	1	3
P_1			3	1	1	2
P_2				2	1	3
P_3					4	5
P_4						3
P_5						

Table 3 Savings in descending order

$s_{ij} = t_{0i} + t_{0j} - t_{ij}$	$s_{ij} = t_{0i} + t_{0j} - t_{ij}$
$s_{12} = 6$	$s_{52} = 4$
$s_{13} = 6$	$s_{24} = 3$
$s_{31} = 6$	$s_{42} = 3$
$s_{15} = 6$	$s_{45} = 1$
$s_{14} = 5$	$s_{54} = 1$
$s_{32} = 4$	$s_{35} = 0$
$s_{25} = 4$	$s_{34} = -1$

Fig. 2 Initial solution

Fig. 3 Second solution



The algorithm starts with 5 routes one for each client. In the next step two of those routes are merged and the solution in Fig. 3 with four routes is obtained. One route with patients 1 and 2, and 3 other routes with the remaining patients. The final solution has only one route that passes by all patients. Note that the restriction not to go from a non-priority patient (blue in Figs. 2 to 6) to a priority patient (red in Figs. 2 to 6) was not broken. Figure 4 shows an example of such an hypothetic intermediate solution that was refused.

The objective of this problem is to minimize the total time of routes (in minutes) visiting all patients. In the initial solution with five routes, the total time of the visit (Z) is 30 min. In the solution obtained in the first iteration, with four routes, Z is 24 min, an improvement of 6 min of the initial solution, whereas the final solution, with only one route, Z is 15 min. The final solution allows a reduction in the total time of the routes of 14 min, saving more than half the time compared to the initial solution (with 5 routes).

Fig. 4 Rejected solution

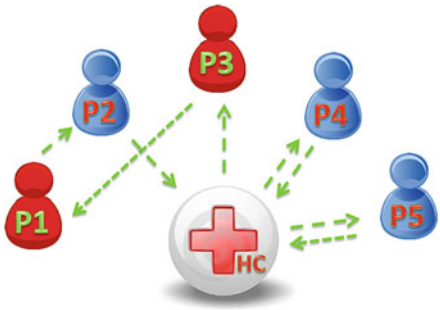


Fig. 5 Third solution

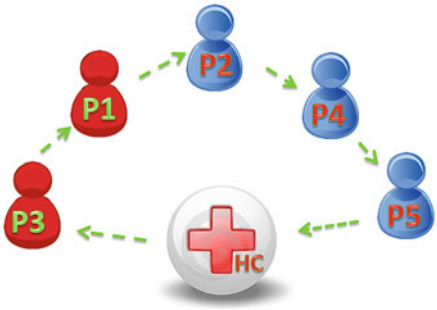


Fig. 6 Final solution

4.3 Algorithm Adaptation

In a first test to the algorithm, with patients from Table 4, it was found that the obtained solutions were not the best ones, although they solved the problem. The algorithm suffers from some myopia because the initial choice of the largest savings does not necessarily lead to the best solution. The total time of the routes obtained for the problem presented in section five was 2 h 28 min. For this time, five routes were obtained: $HC \rightarrow P_1 \rightarrow HC$; $HC \rightarrow P_7 \rightarrow P_2 \rightarrow HC$; $HC \rightarrow P_3 \rightarrow P_8 \rightarrow HC$; $HC \rightarrow P_6 \rightarrow P_4 \rightarrow HC$; and $HC \rightarrow P_5 \rightarrow HC$.

To improve the algorithm some randomness was introduced by imposing jumps in choosing the order of savings, i.e., instead of choosing sequentially in descending

Table 4 Patient data of places to visit (fictitious)

Code	Name	Address	Prioritary	Health procedure
1	José Correia	Rua António Augusto Gonçalves	Yes	Pressure Ulcers
2	Maria Correia	Av. Emídio Navarro 37	No	Oxygen Therapy
3	Rui Fernandes	Rua Coutinhos 26	Yes	Pressure Ulcers
4	Fátima Neves	Estrada da Guarda Inglesa, no 17	No	Ventilation
5	Ana Costa	Rua Caminho das Vinhas 12	Yes	Wound Control
6	Fernando Esteves	Rua da Escola, 37	Yes	Pressure Ulcers
7	Celeste Marques	Av. Cónego Urbano Duarte 92	Yes	Enteral Feeding
8	Mário Gouveia	Rua Augusto Filipe Simões	No	Oxygen Therapy

order savings, the new version of the algorithm skips some of these savings in order to try to find better solutions. This improved version of the algorithm yielded better results than the previous one with no randomness, for the instance tested. The best total time obtained for the routes was 2 h 11 min, with three routes, $HC \rightarrow P_1 \rightarrow P_7 \rightarrow P_2 \rightarrow P_4 \rightarrow HC$, a $HC \rightarrow P_3 \rightarrow P_8 \rightarrow HC$ e $HC \rightarrow P_5 \rightarrow P_6 \rightarrow HC$, which translates into a savings of 17 min. Since this algorithm has a random component better results could be obtained if it was run for longer time. Nevertheless, one of the criteria for choosing the algorithm to be used is how quickly it can get good solutions.

Finally one last approach/adaptation of Clarke and Wright algorithm was tested, which was using a second-order algorithm proposed by [6] and improved by [7]. In this approach, after finding an solution, an arc is fixed and the remaining savings are tested looking for the best solution. When the best solution is found it is saved and the savings that originated are destroyed, except for the one that gave the best solution, which is made permanent. The algorithm runs again considering the arcs that were already made permanent trying to fix more arcs. With this algorithm has achieved a time of 2 h 5 min. The time obtained represents the completion of one route, $HC \rightarrow P_5 \rightarrow P_6 \rightarrow P_1 \rightarrow P_7 \rightarrow P_3 \rightarrow P_8 \rightarrow P_2 \rightarrow P_4 \rightarrow HC$ and allows a saving of 13 min.

5 Development of Health at Home

Given the need to have a central and accessible application to various health care professionals which interacts in the future with the HC information system web based development was chosen. The web application called “Health at Home” was developed using *open source* technologies namely PHP language, Apache Web Server and MySQL database management system which currently belongs to Oracle. Nonetheless the back-end database of the application can be moved to other database management systems like PostgreSQL if needed.

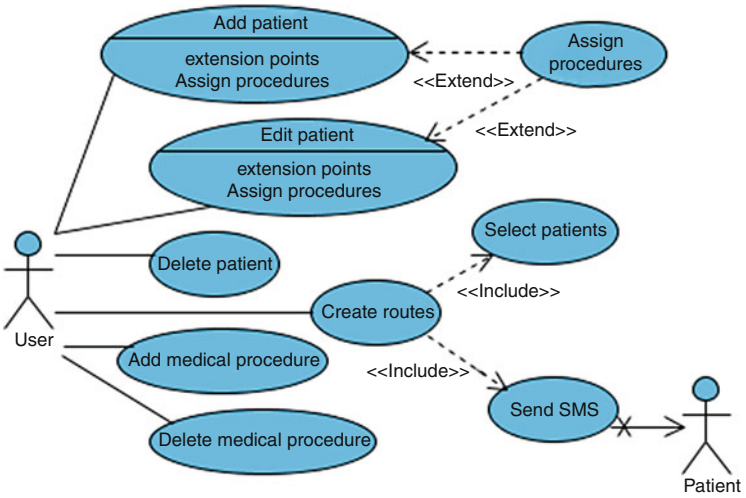


Fig. 7 Use case diagram of the web application

5.1 Functionalities

The web application "Health at Home" will provide the features presented in the use case diagram on Fig. 7.

The user in the diagram of Fig. 7 represents a generic health professional responsible for the management of patients and routes. In "add patients" functionality, data like name, address, whether or not it is a priority patient and procedure/treatment to perform are added. The visits route creation presupposes the choice of patients for whom the health professional wants to create the route and has also the feature of sending SMS messages to patients to inform them of the date and time of visit.

5.2 Presentation of Health at Home

The web application – Health at Home – has three areas of work, one devoted to the patient data management, another one on the management of health procedures (treatments) of patients and a third one to create the routes of home visits, as can see in Fig. 8.

Figure 8 shows the data edition of the fictitious patient José Correia a priority patient. The developed application was for an HC situated in Coimbra, Portugal and that is why the interface is in Portuguese.

Fig. 8 Health at Home (in portuguese – Saúde ao Domicílio)

5.3 Experimentation

To demonstrate the use of the web application – Health at Home – the example from Fig. 1 with 8 patients is shown.

A Table 4 shows the names, addresses, types of patients and health procedures. The patient data is fictitious, although the addresses correspond to actual locations in the city of Coimbra (there is no relation to potential patients in those locations). The only reason to use real addresses is to get the time of travel between locations, which forms the basis for calculations in order to present a demonstration of the implemented algorithm.

The first step is user authentication. Once validated, if you need to create a new patient you may enter the patient management area, otherwise you can go directly to the routes management area.

It is in this area that professional health teams routes are optimized. For this the user must select all patients he wants to add to simulation (in the case of this particular example are the eight patients from Table 4) and run the algorithm, as shown in Fig. 9.

In order to facilitate identification of the type of patient "Health at Home" application marks the priority patients with a red bullet and the non-priority patients with a green bullet.

Figures 10 and 11 show the obtained result in a list and map visualizations.

In this case the obtained result is only one route, $HC \rightarrow 5 \rightarrow 6 \rightarrow 1 \rightarrow 7 \rightarrow 3 \rightarrow 8 \rightarrow 2 \rightarrow 4 \rightarrow HC$, which appears in map visualization (Fig. 11) in alphabetic order.

The solutions presented include the time of travel between patients' homes (arcs) and time spent at the patients' homes (nodes). Although it could have

Fig. 9 Health at Home – patient selection for route creation

Trajecto da 1 ª equipa:
Saída do Centro de Saúde:
5 - Rua Caminho das Vinhas 12
6 - Rua da Escola, 37, 3040-563 Coimbra
1 - Rua António Augusto Gonçalves, 3041-901 Coimbra
7 - Avenida Cónego Urbano Duarte 92, 3030 Coimbra
3 - Rua Coutinhos 26, 3000 Coimbra
8 - Rua Augusto Filipe Simões, 3000 Coimbra
2 - Avenida Emídio Navarro 37, 3000-150 Coimbra
4 - Estrada da Guarda Inglesa, N.º 17, 3040-193 Coimbra
Regresso ao Centro de Saúde!

O TEMPO FINAL DESTA SOLUÇÃO É: 02:05:00

Fig. 10 Health at Home – list visualization of proposed route

considered different times for demonstration purpose it was considered that the medical teams spend 10 min at a patient's home to perform the respective health procedure (treatment).

6 Final Remarks

The proposed application optimizes the routes of professional health teams' visits, which has the effect of reducing fuel costs and improving the efficiency of the system.

The innovation of this application, which is not found in the literature review, is to optimize the routes of visits of professional health teams accordingly to the asepsis degree of patients, allowing their differentiation into priority and non-priority, thus helping avoiding the transmission of infections between patients.

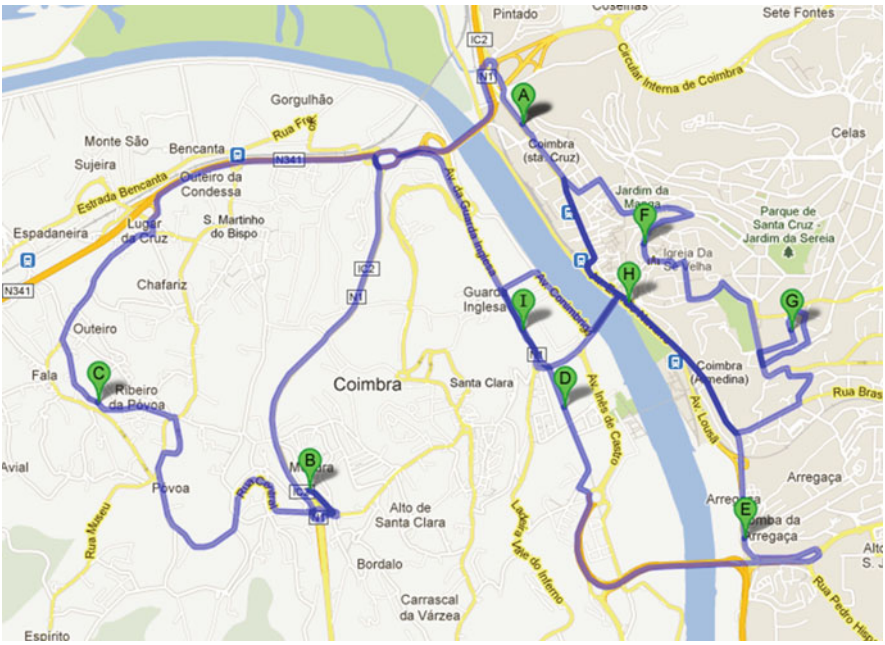


Fig. 11 Health at Home – map visualization of proposed route

Future research will focus on identifying additional constraints, such as the times when the patients would prefer to be visited or preference of patients for certain health professionals [3] in order to improve the satisfaction of patients and health professionals in the service.

Another situation not addressed, but also unsolicited, was the issue of optimizing the composition of teams of health professionals referenced in other similar works [3, 10] and that may be the subject of study in future developments. The currently implemented algorithm sees teams as single entities regardless of the number of elements that compose them.

The next step will be to create a schedule that allows the application to list patients who need to be visited on a particular day.

Notwithstanding the work developed and presented in this article has been carried out on the basis of HC in an urban area (city of Coimbra), which can be generalized to any other geographic location.

Acknowledgements We would like to thank all help given by our Information Technology Management course colleagues that have contributed for the development of this project, in particular to Daniel Fernandes that contributed with advises to some design issues of the web application. We would also like to thank nurse Mariana Costa, a nurse practitioner, for the idea that originated this entire project, as well for her support. Lastly we would like to thank the reviewers for the corrections and the suggestions that helped improve this paper.

References

1. Braysy, O., Nakari, P., Dullaert, W., Neittaanmaki, P.: An optimization approach for communal home meal delivery service: a case study. J. Comput. Appl. Math. (2009). doi:10.1016/j.cam.2008.10.038
2. Benzarti, E., Sahin E., Dallery Y.: Operations management applied to home care services: analysis of the districting problem. Decis. Support Syst. (2012). doi:10.1016/j.dss.2012.10.015
3. Bertels, S., Fahle T.: A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. Comput. Oper. Res. (2006). doi:10.1016/j.cor.2005.01.015
4. Clarke, G., Wright, J.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
5. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. Eur. J. Oper. Res. (1992). doi:10.1016/0377-2217(92)90192-C
6. Karnaugh M.: A new class of algorithms for multipoint network optimization. IEEE Trans. Commun. (1976). doi:10.1109/TCOM.1976.1093334
7. Martins, P.: Enhanced second order algorithm applied to the capacitated minimum spanning tree problem. Comput. Oper. Res. (2007). doi: 10.1016/j.cor.2005.09.017
8. Ministério da Saúde.: Cuidados Continuados Integrados nos CSP – Carteira de Serviços. Ministério da Saúde (2007)
9. Moriya, T., Módena, J.L.P.: Assepsia e antissepsia: técnicas de esterilização. Simpósio: Fundamentos em Clínica Cirúrgica – 1ª Parte **41**(3), 265–273 (2008)
10. Nickel, S., Schröder, M., Steeg, J.: Mid-term and short-term planning support for home health care services. Eur. J. Oper. Res. (2012). doi:10.1016/j.ejor.2011.10.042
11. Nunes, J., Matos, L., Trigo, A.: Taxi pick-ups route optimization using genetic algorithms. In: Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA'11) – Volume Part I, Ljubljana (2011). doi:10.1007/978-3-642-20282-7_42
12. Rasmussen, M.S., Justesen, T., Dohn, A., Larsen J.: The home care Crew scheduling problem: preference-based visit clustering and temporal dependencies. Eur. J. Oper. Res. (2012). doi:10.1016/j.ejor.2011.10.048

Cell-Free Layer Measurements in a Network with Bifurcating Microchannels Using a Global Approach

David Bento, Diana Pinho, Ana I. Pereira, and Rui Lima

Abstract One of the most interesting hemodynamic phenomenon observed in microchannels is the existence of a marginal cell-free layer (CFL) at regions adjacent to the wall. This is a well known phenomenon that occurs in simple glass capillaries and in vivo microvessels, but has never been investigated in detail in biomedical microdevices containing complex geometries. In the present chapter, in vitro blood flowing through bifurcating microchannels was studied, with the aim of characterizing the cell-free layer (CFL). For that three different videos with different hematocrit and flow rates were considered. All images were obtained by means of a high-speed video microscopy system and then processed in MatLab using the Image Processing toolbox. The numerical data was obtained automatically and analyzed by optimization techniques using the genetic algorithm approach. The results suggest that the CFL were formed in a similar way at the upper and lower regions in all bifurcations.

1 Introduction

Blood is a complex biofluid, composed mainly of red blood cells (RBCs) and plasma, which contains a massive amount of information about several physiological and pathologic events happening throughout the human body. Hence, in

D. Bento (✉) • D. Pinho
Polytechnic Institute of Bragança, Bragança, Portugal

CEFT, Faculty of Engineering at University of Porto, Porto, Portugal
e-mail: davidbento@ipb.pt; diana@ipb.pt

A.I. Pereira
Polytechnic Institute of Bragança, Bragança, Portugal

ALGORITMI, University of Minho, Guimarães, Portugal
e-mail: apereira@ipb.pt

R. Lima
Mechanical Engineering Department, University of Minho, Guimarães, Portugal

Polytechnic Institute of Bragança, Bragança, Portugal

CEFT, Faculty of Engineering at University of Porto, Porto, Portugal
e-mail: ruimec@ipb.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_4

vitro blood studies in microfluidic devices have been intensively performed in order to obtain a better understanding on the blood flow behavior at microscale levels [14]. A hemodynamic phenomenon observed in in vitro studies has shown that in microchannels both hematocrit (Hct) and apparent blood viscosity decrease as the tube diameter is reduced [10, 14, 21]. This phenomenon mainly causes the formation of a cell-free layer (CFL) around the wall, which is related with the tendency of the RBCs to migrate toward the center of the microchannel [4]. Recently several studies showed strong evidence that the formation of the CFL is affected by both microchannel geometry [13, 17, 20, 22] and physiological conditions of the working fluid [7, 8].

Although there have been several studies on the measurement of CFL thickness in simple geometries [12, 16], according to our knowledge there have been few studies on the measurement of the CFL in complex geometries, such as a network containing multiple bifurcations and confluences [2, 3]. To study the behavior of the CFL along complex geometries, image analysis plays an important role [18, 19].

Image analysis processing is a huge area which provides a large number of viable applications. Segmentation is one of the most important elements in automated image analysis, mainly because at this step the objects or other entities of interest are extracted from the original image for subsequent processing, such as description and recognition [1]. A variety of techniques can be applied: simple methods as thresholding, or complex methods such as edge/boundary detection or region growing. The literature contains hundreds of segmentation techniques [6], but there is no single method that can be considered good enough for all kinds of images. In this work an automatic method able to measure the thickness of the CFL in different areas of the microchannel was developed. The automatic method consists in the application of preprocessing filters, to remove and smooth the noise, followed by the sum of the difference and multiplication of consecutive frames.

As well as image processing, optimization has been showing its important role in this area of study such as in microcirculation and its phenomena. In the recent years, the population based algorithms have become increasingly robust and easy to use [9]. These algorithms are based in the Darwin theory of evolution, performing a search for best solution among a population that evolves through several generations. The genetic algorithm (GA), introduced by John Holland [11] and popularized by his student David Goldberg in the late 1980s [9], is a search method based on population genetics, inspired by the Darwinian principle of natural selection and genetic replication, in which the principle of selection favors the most fit individual with longer life and therefore more likely to reproduce. Additionally individuals with more offsprings have more opportunity to perpetuate their genetic to the subsequent generations [9, 11]. GAs are able to search and find the global minimum, but have a high computational cost, as result of the need of evaluating the total population at each iteration.

Generally the GAs use three main types of rules at each step to create the next generation from the current population. Briefly, it selects individuals from the current population (named as parents), that contribute with children to the

population at the next generation. Children could be generated by combination of two parents (crossover) or by random changes in the individual parents (mutation).

The GA differs from classical optimization algorithms in two main ways. The classical optimization algorithms generate either a population of points at each iteration, and the best point in the population approaches an optimal solution, or a single point at each iteration, and the sequence of points approaches an optimal solution. However the GA selects the next population by computation which uses random number generators instead of selecting the next point in the sequence by a deterministic computation as in the classical optimization algorithms [15].

In this work we used the genetic algorithm included in the Global Optimization toolbox from Matlab and applied it to the experimental data obtained by an automatic method developed to measure the CFL in microchannels.

The main purpose of this chapter is not only to measure the CFL in a network with bifurcating microchannels at different working fluids, by means of an automatic method developed in MatLab [2, 3], but also to characterize the CFL along the network using global optimization techniques.

This chapter is organized as follows. Section 2 presents the materials used in this work and the methods that were applied in this study, in particular the experimental set-up, the working fluids, the image processing techniques and the optimization method. Section 3 presents the numerical results and some discussion. The last section presents some conclusions and future work.

2 Materials and Methods

2.1 Experimental Set-Up

The high-speed video microscopy system used in this study consists of an inverted microscope (IX71; Olympus) combined with a high-speed camera (i-SPEED LT). The microchannel was placed on the stage of the inverted microscope and, by using a syringe pump (PHD ULTRA), a pressure-driven flow was kept constant (cf. Fig. 1).

The series of microscope images were captured with a resolution of 600×800 pixels. All the images were recorded at the center plane of the microchannels at a rate of 200 frames/second and were transferred to the computer and then evaluated using image analysis techniques.

2.2 Working Fluids and Microchannel Geometry

The blood samples were collected from a healthy adult sheep, and ethylenediaminetetraacetic acid (EDTA) was added to prevent coagulation. The red blood cells (RBCs) were separated from the blood by centrifugation and washed twice with physiological saline (PS). The washed RBCs were suspended in Dextran 40 (Dx 40)

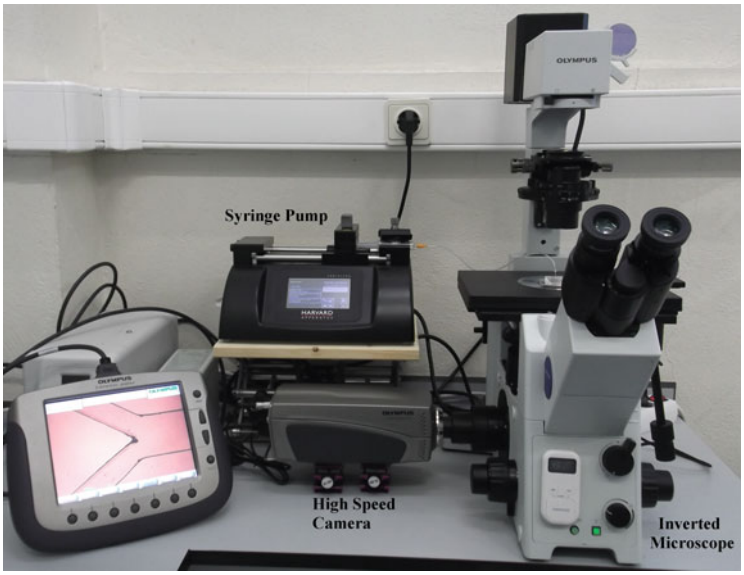


Fig. 1 High-speed video microscopy system used in this study

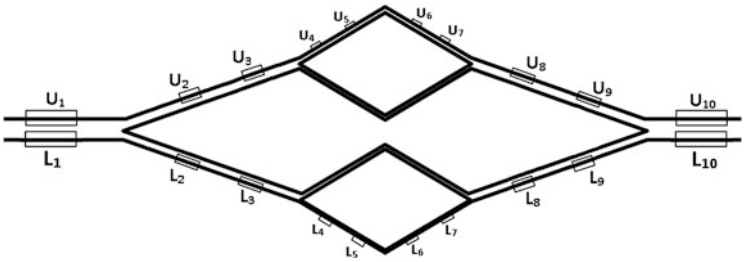


Fig. 2 The geometry of the network and the regions where the CFL was measured

to make up the required RBCs concentration by volume – hematocrit. In this study the Hct of 5 % and 10 % were used. All blood samples were stored hermetically at 4 °C until the experiment was performed.

The microchannels fabricated for the proposed study have been produced in polydimensiloxane (PDMS) by a soft-lithography technique. The geometry used is a network of microchannels, containing several bifurcations and confluences. Figure 2 illustrates the configuration of the network and the regions where the CFL was measured.

The region U_i corresponds to the upper cell-free layer, for $i = 1, \ldots, 10$ and L_i the lower cell-free layer, for $i = 1, \ldots, 10$.

The flow rate used for the recorded videos was 1000 nl/min for the both levels of Hct (5 % and 10 %) and 500 nl/min only for the Hct of 10 %.

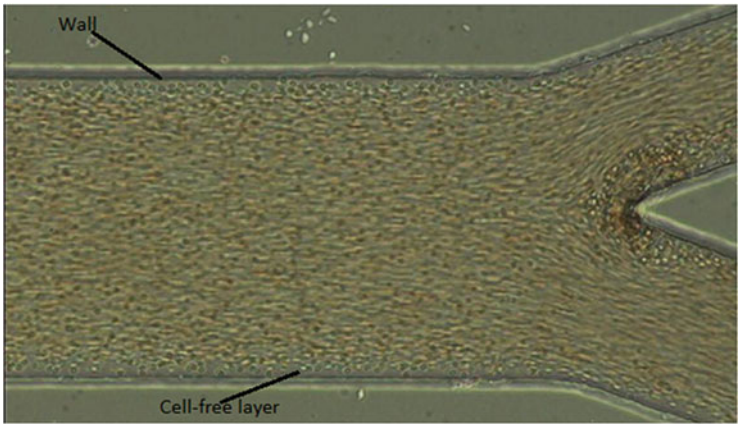


Fig. 3 The original image of blood sample flowing in a bifurcating microchannel

2.3 Image Analysis

Figure 3 shows an example of the image sequences studied in this work, which were processed using the Image Processing toolbox, available in MatLab [6]. An automatic method to measure the CFL was developed and tested.

The method consists in the combination of the binarization of the sum of the difference between the consecutive frames and the sum of the multiplication of the image sequence. The general steps of the method are:

- Preprocess the image to smooth and eliminate the artifacts;
- Obtain the difference between consecutive frames and sum;
- Binarize the sum image;
- Apply the multiplication of consecutive frames from original image sequence and sum;
- Multiply the last image with the sum of the differences image;
- Select the area to obtain the required data.

Firstly a median filter with a 3×3 pixel mask was applied to each frame to reduce the noise, and then the difference of the consecutive frames was computed and those differences are summed. An image that represents the sum of the differences of all images was obtained. The resulting image is binarized, yielding an region in which black is the region of highest intensity of RBCs, as is possible to observe in Fig. 4.

In the second step, the original image sequence was analyzed once again to obtained an image that represents the sum of the multiplication of the frames, where the channel is the white region. As the third step, those images were multiplied by the images obtained in the previous step Fig. 4. Figure 5 shows the result image where the white region is the region of cell-free layer.

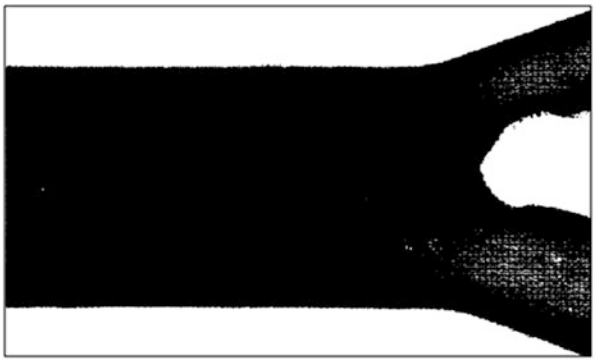


Fig. 4 Result image of the sum of the differences

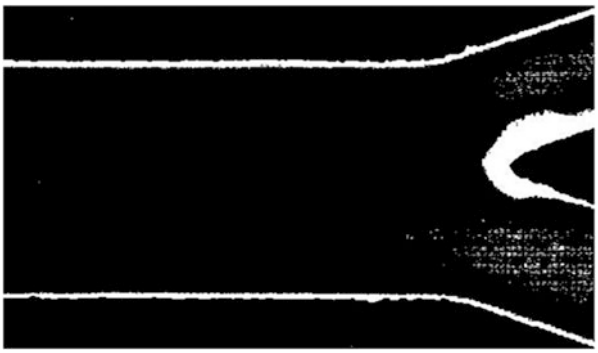


Fig. 5 Result image that shows in white boundary of CFL

Finally, the region of interest was selected and the upper and lower CFLs were automatically measured.

2.4 Mathematical Model

The main objective is to compare the behavior of the cell-free layer in fluids with different Hcts and with difference flow rates. To accomplish that the nonlinear least squares theory was used.

In each region U_i, for $i = 1, \ldots, 10$, and L_i, for $i = 1, \ldots, 10$, the nonlinear optimization problem defined bellow was considered

$$\min_{y \in \mathbb{R}^n} f(y) \equiv \sum_{k=1}^{N_R} (M_k - g_h(y, x_k))^2$$
$$s.t \quad g_h(y, x_k) \geq 0 \ \forall k = 1, \ldots, N_R \tag{1}$$

where (M_k, x_k), for $k = 1, \ldots, N_R$ are the CFL measurements of the region R (defined as U_i and L_i, for $i = 1, \ldots, 10$) and n is the number of the variables y of each function g_h. The functions g_h, for $h = 1, \ldots, 3$, are defined as follows

$$\begin{aligned} g_1(y, x) &= y_1 x^2 + y_2 x + y_3, \\ g_2(y, x) &= y_1 x + y_2, \\ g_3(y, x) &= \sin(y_1 x) + \cos(y_2 x) + y_3. \end{aligned} \tag{2}$$

The functions g_1, g_2 and g_3 were proposed in previous works presented in [17–19].

2.5 Global Optimization Method: Genetic Algorithm

In the present work we used the genetic algorithm (GA), which is an optimization technique based on the evolution principles. The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be "parents" and uses them to produce the "children" for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

The evolution can be obtained by a crossover process, where the genes of the best individuals are crossed with genes from other individuals which also have good performance. The selection of the genes is done randomly. The GA also applies the concept of mutation, thus improving the optimization process by, randomly, introduction values that were not present in the previous generations.

The GA can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. The GA can address problems of mixed integer programming, where some components are restricted to be integer-valued [9].

The Genetic Algorithms, are heuristic procedures, and does not guarantee that the global minimum is found, however it is accepted that the final solution is close to the global minimum, after a sufficient number of iterations [5].

The following outline summarizes how the genetic algorithm works:

1. The algorithm begins by creating a random initial population.
2. A sequence of populations is created by using the individuals in the current generation to create the next population. For generation the new population, the algorithm performs the following steps:
   - Scores each member of the current population by computing its fitness value.
   - Scales the raw fitness scores to convert them into a more usable range of values.

- Randomly select members, called parents.
- Some of the individuals in the current population that have higher fitness are chosen as elite. These elite individuals are passed to the next population.
- Produces children from the parents. Children are produced either by making random changes to a single parent – mutation – or by combining the vector entries of a pair of parents – crossover.
- Replaces the current population with the children and the elite individuals.

3. The algorithm stops when one of the stopping criteria is met.

The iterative procedure terminates when there is no significant difference between two successive populations, i.e., the corresponding difference is smaller than ϵ.

3 Results and Discussion

The numerical results were obtained using a Intel(R) Core (TM) i3 CPU M330@2.13 GHz with 8.00 GB of RAM. The captured videos were analyzed and the numerical data was taken in the regions already defined in Fig. 2. Three different flow condition were studied:

- fluid with a flow rate 500 nl/min and with 10 % of Hct,
- fluid with a flow rate 1000 nl/min and with 10 % of Hct, and
- fluid with a flow rate 1000 nl/min and with 5 % of Hct.

The following tables present the obtained numerical results using the genetic algorithm to solve the optimization problem (1). Since the genetic algorithm is a stochastic method, each problem was solved 30 times, considering different initializations with random populations. Table 1 presents the regions where problem (1) was solved, the average of the optimum value and the minimum value obtained in the 30 runs.

The results from Table 1 show that the best curve fit is g_3, since the minimum value appears more often for this function (14 out of 20 regions). The function g_1 has obtained the best value in the region U_2, U_3 and L_{10}. In some regions, namely L_2, L_3 and L_8, it was not possible to conclude anything about the best fit, since the average of the optimum value was obtained with a given function but the minimum of all runs was obtained with another curve fitting. Nevertheless, the difference between the results obtained by the two functions, g_1 and g_3 was not significative.

Table 2 presents the results obtained using the GA for the fluid with a flow rate of 1000 nl/min and Hct of 5 %. With these conditions the function that better fits the experimental data was g_3. However there were some regions as U_1, U_2, U_4, U_8 and U_9 where it is not possible to conclude about the best fit.

It is worth mentioning that in all lower regions the best fit is the g_3 function and the function g_2 presents always the worst results.

Table 1 Numerical results obtained for 10 % of Hct at the flow rate of 500 nl/min

	Upper cell-free layer				Lower cell-free layer		
Region	Function	Average	Minimum	Region	Function	Average	Minimum
U_1	g_1	1.629E+03	1.118E+03	L_1	g_1	1.293E+03	8.738E+02
	g_2	3.809E+07	2.504E+04		g_2	6.220E+07	1.103E+04
	g_3	**1.190E + 03**	**9.235E + 02**		**g_3**	**1.005E + 03**	**8.050E + 02**
U_2	**g_1**	**2.453E + 03**	**2.179E + 03**	L_2	g_1	2.087E+03	**1.503E+03**
	g_2	3.572E+07	1.493E+05		g_2	2.534E+08	1.411E+05
	g_3	2.695E+03	2.210E+03		g_3	**2.058E + 03**	1.570E+03
U_3	g_1	1.387E+03	**9.423E + 02**	L_3	g_1	**2.305E + 03**	$2.236E + 03$
	g_2	1.229E+08	1.268E+03		g_2	1.956E+07	4.110E+03
	g_3	**1.184E + 03**	1.001E+03		g_3	2.786E+03	**1.967E+03**
U_4	g_1	2.309E+03	1.216E+03	L_4	g_1	1.367E+03	1.300E+03
	g_2	1.151E+07	6.819E+04		g_2	1.251E+06	4.277E+03
	g_3	**1.087E + 03**	**8.464E + 02**		**g_3**	**1.054E + 03**	**8.864E + 02**
U_5	g_1	1.183E+03	1.007E+03	L_5	g_1	1.884E+03	1.588E+03
	g_2	1.451E+07	2.775E+03		g_2	8.912E+05	3.241E+03
	g_3	**1.247E + 00**	**9.906E + 02**		**g_3**	**1.461E + 03**	**1.338E + 03**
U_6	g_1	8.272E+02	7.151E+02	L_6	g_1	1.262E+03	1.135E+03
	g_2	1.987E+06	6.934E+02		g_2	1.656E+07	8.391E+03
	g_3	**6.088E + 02**	**5.226E + 02**		**g_3**	**9.821E + 02**	**7.883E + 02**
U_7	g_1	1.686E+03	1.107E+03	L_7	g_1	2.435E+03	2.312E+03
	g_2	3.380E+07	1.427E+03		g_2	3.886E+07	6.669E+04
	g_3	**1.274E + 03**	**9.526E + 02**		**g_3**	**1.994E + 03**	**1.524E + 03**
U_8	g_1	5.352E+03	4.230E+03	L_8	g_1	8.464E+02	**5.691E+02**
	g_2	8.379E+07	1.056E+05		g_2	2.748E+07	1.738E+05
	g_3	**4.541E + 03**	**3.803E + 03**		g_3	**7.490E + 02**	6.324E+02
U_9	g_1	2.361E+03	1.979E+03	L_9	g_1	2.199E+03	1.326E+03
	g_2	3.153E+07	8.790E+03		g_2	1.865E+08	1.512E+03
	g_3	**1.959E + 03**	**1.606E + 03**		**g_3**	**1.598E + 03**	**1.264E + 03**
U_{10}	g_1	2.329E+03	1.680E+03	L_{10}	**g_1**	**1.116E + 03**	**8.477E + 02**
	g_2	7.132E+06	3.544E+03		g_2	2.708E+08	3.206E+05
	g_3	**1.829E + 03**	**1.219E + 03**		g_3	1.946E+03	1.320E+03

Table 3 presents the data obtained with a genetic algorithm for the fluid with a flow rate of 1000 nl/min and 10 % of Hct.

Overall, the best results were for the fluid having a flow rate of 1000 nl/min and 10 % of Hct, as all tested functions present the minimum results when compared with the values obtained in Tables 1 and 2.

Additionally, the function g_3 is the one that presents the best fit to the data. In contrast the function g_2 is the one that presents the worst results. In the region L_7 it was the function g_1 that has shown the best fit. In the regions U_2, U_3, U_4, U_7, U_8 and U_9 it was not possible to conclude anything about the best fitting.

Table 2 Numerical results obtained for 5 % of Hct at the flow rate of 1000 nl/min

	Upper cell-free layer				Lower cell-free layer		
Region	Function	Average	Minimum	Region	Function	Average	Minimum
U_1	g_1	2.247E+03	**1.735E+03**	L_1	g_1	2.243E+03	2.123E+03
	g_2	3.483E+07	2.798E+03		g_2	3.584E+07	3.061E+03
	g_3	**1.985E+03**	$1.750E+03$		**g_3**	**1.969E + 03**	**1.488E + 03**
U_2	g_1	6.018E+02	**4.062E+02**	L_2	g_1	1.379E+04	6.316E+03
	g_2	2.119E+07	3.224E+03		g_2	4.980E+07	2.276E+05
	g_3	**5.016E + 02**	$4.391E+02$		**g_3**	**1.202E + 04**	**3.577E + 03**
U_3	g_1	1.458E+03	7.895E+02	L_3	g_1	3.182E+03	2.822E+03
	g_2	7.733E+07	1.043E+04		g_2	1.189E+07	2.367E+03
	g_3	**9.168E + 02**	**7.541E + 02**		**g_3**	**2.486E + 03**	**2.011E + 03**
U_4	g_1	4.254E+02	**3.189E+02**	L_4	g_1	1.617E+03	1.394E+03
	g_2	6.115E+06	1.309E+04		g_2	1.420E+06	1.367E+03
	g_3	**3.937E + 02**	$3.354E+02$		**g_3**	**1.375E + 03**	**1.001E + 03**
U_5	g_1	1.206E+03	9.817E+02	L_5	g_1	4.840E+02	4.454E+02
	g_2	1.110E+07	6.293E+03		g_2	1.238E+05	4.012E+02
	g_3	**1.045E + 03**	**8.344E + 02**		**g_3**	**3.848E + 02**	**3.130E + 02**
U_6	g_1	1.406E+03	9.314E+02	L_6	g_1	2.508E+03	1.861E+03
	g_2	1.866E+06	2.345E+03		g_2	4.331E+06	3.067E+03
	g_3	**9.992E + 02**	**7.712E + 02**		**g_3**	**1.732E + 03**	**1.507E + 03**
U_7	g_1	8.085E+02	6.668E+02	L_7	g_1	3.112E+03	2.834E+03
	g_2	1.768E+07	3.206E+04		g_2	3.775E+07	1.219E+04
	g_3	**7.813E + 02**	**6.314E + 02**		**g_3**	**2.699E + 03**	**2.314E + 03**
U_8	g_1	**1.480E + 03**	$1.331E+03$	L_8	g_1	3.232E+03	2.888E+03
	g_2	1.593E+07	2.499E+03		g_2	3.653E+07	1.357E+04
	g_3	1.779E+03	**1.204E+03**		**g_3**	**3.132E + 03**	**2.834E + 03**
U_9	g_1	**2.859E + 03**	$2.647E+03$	L_9	g_1	4.070E+03	3.251E+03
	g_2	3.768E+07	3.003E+04		g_2	2.127E+08	5.987E+04
	g_3	3.012E+03	**2.601E+03**		**g_3**	**3.400E + 03**	**2.742E + 03**
U_{10}	g_1	7.451E+03	6.493E+03	L_{10}	g_1	8.249E+02	4.447E+02
	g_2	2.365E+07	1.626E+04		g_2	1.578E+07	6.985E+03
	g_3	**6.359E + 03**	**5.190E + 03**		**g_3**	**5.229E + 02**	**4.270E + 02**

When comparing the three fluids one possible conclusion is that in the lower regions the best curve fit is based on the sum of trigonometric functions.

Regarding to the results presented in the Table 4 resumes the previous tables by presenting the best fit for each combination of region and fluid. Notations $F_{10,500}$, $F_{5,1000}$ and $F_{10,1000}$ are use to represent fluid with flow rate of 1000 nl/min and 10 % of Hct, fluid with flow rate of 500 nl/min and 10 % and fluid with flow rate of 1000 nl/min and 5 % of Hct, respectively.

Overall, it is possible to conclude that at lower regions the CFL behaves in similar way, i.e., as a sum of trigonometric functions.

Table 3 Numerical results obtained for 10 % of Hct at the flow rate of 1000 nl/min

	Upper-free layer				Lower cell-free layer		
Region	Function	Average	Minimum	Region	Function	Average	Minimum
U_1	g_1	1.237E+03	9.064E+02	L_1	g_1	1.999E+03	1.228E+03
	g_2	1.672E+07	5.643E+03		g_2	6.973E+07	3.581E+03
	$\mathbf{g_3}$	**1.058E + 03**	**7.780E + 02**		$\mathbf{g_3}$	**1.430E + 03**	**1.089E + 03**
U_2	g_1	5.388E+02	**3.685E+02**	L_2	g_1	5.884E+02	4.500E+02
	g_2	3.030E+07	1.465E+04		g_2	1.280E+07	4.957E+02
	g_3	**5.264E + 02**	4.106E + 02		$\mathbf{g_3}$	**4.871E + 02**	**3.968E + 02**
U_3	g_1	6.419E+02	**4.153E+02**	L_3	g_1	1.195E+03	7.888E+02
	g_2	2.440E+07	3.165E+04		g_2	1.110E+07	1.006E+03
	g_3	**5.830E + 02**	4.809E + 02		$\mathbf{g_3}$	**9.081E + 02**	**6.591E + 02**
U_4	g_1	6.632E+02	**3.059E+02**	L_4	g_1	6.509E+03	5.031E+03
	g_2	5.379E+06	1.211E+04		g_2	5.467E+06	1.325E+04
	g_3	**3.836E + 02**	3.233E + 02		$\mathbf{g_3}$	**5.411E + 03**	**2.868E + 03**
U_5	g_1	1.181E+03	1.037E+03	L_5	g_1	8.213E+02	6.233E+02
	g_2	2.315E+07	3.016E+03		g_2	1.480E+06	8.988E+02
	$\mathbf{g_3}$	**1.075E + 03**	**8.056E + 02**		$\mathbf{g_3}$	**7.997E + 02**	**5.619E + 02**
U_6	g_1	1.701E+03	1.515E+03	L_6	g_1	1.003E+03	9.235E+02
	g_2	8.977E+06	2.961E+03		g_2	6.573E+06	2.848E+03
	$\mathbf{g_3}$	**1.689E + 03**	**1.287E + 03**		$\mathbf{g_3}$	**9.809E + 02**	**8.705E + 02**
U_7	g_1	7.789E+02	**4.139E+02**	L_7	$\mathbf{g_1}$	**2.491E + 03**	**2.189E + 03**
	g_2	2.122E+07	1.340E+04		g_2	2.028E+08	8.873E+03
	g_3	**5.957E+02**	4.738E + 02		g_3	2.667E+03	2.211E+03
U_8	g_1	**3.338E + 03**	2.935E + 03	L_8	g_1	6.292E+03	2.258E+03
	g_2	7.077E+07	8.235E+04		g_2	5.678E+07	3.609E+03
	g_3	3.424E+03	**2.709E+03**		$\mathbf{g_3}$	**2.119E + 03**	**1.740E + 03**
U_9	g_1	9.113E+02	**5.342E+02**	L_9	g_1	1.789E+03	1.323E+03
	g_2	1.027E+08	1.593E+04		g_2	2.086E+08	3.922E+04
	g_3	**7.879E + 02**	5.786E + 02		$\mathbf{g_3}$	**1.432E + 03**	**1.221E + 03**
U_{10}	g_1	5.416E+02	3.296E+02	L_{10}	g_1	5.578E+02	4.096E+02
	g_2	6.169E+06	5.394E+02		g_2	1.206E+07	6.349E+03
	$\mathbf{g_3}$	**4.249E + 02**	**2.643E + 02**		$\mathbf{g_3}$	**5.033E + 02**	**3.368E + 02**

Another interesting result is that the regions U_5 and U_6 have similar behavior when compared with L_5 and L_6. This behavior happens in the smaller microchannels and as a result we may conclude that the width of the channels may influence the quality of the images and consequently the image analysis results.

Overall our results show that the CFLs have a similar behavior in all bifurcations. However, for the upper regions it is not so clear to conclude about the best fit function.

More investigation is required in this area and in the mean time, we plan to measure the CFL by using a manual tracking method and compare the results with the automatic method applied in this study.

Table 4 Best fit for all considered fluids

	Upper-free layer				Lower cell-free layer		
Region	$F_{10,500}$	$F_{5,1000}$	$F_{10,1000}$	Region	$F_{10,500}$	$F_{5,1000}$	$F_{10,1000}$
U_1	g_3	–	g_3	L_1	g_3	g_3	g_3
U_2	g_1	–	–	L_2	–	g_3	g_3
U_3	g_1	g_3	–	L_3	–	g_3	g_3
U_4	g_3	–	–	L_4	g_3	g_3	g_3
U_5	g_3	g_3	g_3	L_5	g_3	g_3	g_3
U_6	g_3	g_3	g_3	L_6	g_3	g_3	g_3
U_7	g_3	g_3	–	L_7	g_3	g_3	g_1
U_8	g_3	–	–	L_8	–	g_3	g_3
U_9	g_3	–	–	L_9	g_3	g_3	g_3
U_{10}	g_3	g_3	g_3	L_{10}	g_1	g_3	g_3

4 Conclusions and Future Work

In this study, we presented an automatic image processing method to obtain automatically the CFL measurements in a complex microchannel with bifurcation and confluence. The CFL boundary was fit using three different functions and a genetic algorithm was used to solve the constrained optimization problem. The best fit was obtained using the function g_3, i.e. a sum of trigonometric functions. As future work, we will test more fluids with different properties and different functions to fit the CFL measurements and compare the automatic results obtained in this work with a manual tracking method.

Acknowledgements The authors acknowledge the financial support provided by PTDC/SAU-ENB/116929/2010, EXPL/EMS-SIS/2215/2013 and scholarships SFRH/BD/89077/2012 and SFRH/BD/91192/2012 from FCT (Science and Technology Foundation), COMPETE, QREN and European Union (FEDER).

References

1. Acharya, T., Ray, A.K.: Image Processing Principles and Applications. Wiley, Hoboken (2005)
2. Bento, D., Pinho, D., Pereira, A., Lima, R.: Cell-free layer (CFL) analysis in a glass capillary: comparison between a manual and automatic method. AIP Conf. Proc. **1479**, 786–789 (2012)
3. Bento, D., Pinho, D., Pinto, E., Tomoko, Y., Correia, T., Lima, J., Pereira, A.I., Lima, R.: Cell-free layer measurements in a bifurcation microchannel: comparison between a manual and automatic methods. In: 5th Portuguese Congress on Biomechanics, Espinho, pp. 359–362 (2013)
4. Caro, C., Pedley, T., Schroter, R., Seed, W.: The Mechanics of the Circulation. Oxford University Press, Oxford/New York (1978)
5. Catlin, G., Advani, S., Prasad, A.: Optimization of polymer electrolyte membrane fuel cell channels using a genetic algorithm. J. Power Sources **196**, 9407–9418 (2011)

6. Eddins, S.L., Gonzalez, R.C., Woods, R.E.: Digital Image Processing Using Matlab. MathWorks. Gatesmark Publishing, New York (2002)
7. Fujiwara, H., Ishikawa, T., Lima, R., Matsuki, N., Imai, Y., Kaji, H., Nishizawa, M., Yamaguchi, T.: Red blood cells motions in high-hematocrit blood flowing through a stenosed microchannel. J. Biomech. **42**, 838–843 (2009)
8. Garcia, V., Dias, R., Lima, R.: In vitro blood flow behaviour in microchannels with simple and complex geometries. In: Naik, G.R. (ed.) Applied Biological Engineering – Principles and Practice, vol. 17, pp. 394–416. InTechcs, Rijeka (2012). J. Biomech.
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
10. Goldsmith, H., Cokelet, G., Gaehtgens, P.: Robin Fahraeus: evolution of his concepts in cardiovascular physiology. Am. J. Physiol. **257**, H1005–H10015 (1989)
11. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
12. Kim, S., Kong, R.L., Popel, A.S., Intaglietta, M., Jonhson J.C.: A computer-based method for determination of the cell-free layer width in microcirculation. Microcirculation **13**, 199–207 (2006)
13. Leble, V., Lima, R., Dias, R.P., Fernandes, C.S., Ishikawa, T., Imai, Y., Yamaguchi, T.: Asymmetry of red blood cell motions in a microchannel with a diverging and converging bifurcation. Biomicrofluidics **5**, 044120-1–044120-15 (2011)
14. Lima, R., Ishikawa, T., Imai, Y., Yamaguchi, T., Dias et al.: Single and Two-Phase Flows on Chemical and Biomedical Engineering, pp. 513–547. Bentham Science, Sharjah (2012)
15. Matlab: Global Optimization Toolbox. MathWorks (2012)
16. Namgung, B., Ong, P.K., Wong, Y.H., Lim, D., Chun, K.C., Kim, S.: A comparative study of histogram-based thresholding methods for the determination of cell-free layer width in small blood vessels. Physiol. Meas. **31**, N61–N70 (2010)
17. Pinho, D.: Determination and characterization of red blood cells trajectories: a semi-automatic method. Master in biomedical technology. Polytechnic Institute of Bragança (2011, in portuguese)
18. Pinho, D., Lima, R., Pereira, A.I., Gayubo, F.: Automatic tracking of labeled red blood cells in microchannels. Int. J. Numer. Methods Biomed. Eng. **29**(9), 977–987 (2013)
19. Pinho, D., Lima, R., Pereira, A.I., Gayubo, F.: Tracking red blood cells in microchannels: a comparative study between an automatic and a manual method. In: Tavares, J.M.R.S., Jorge, R.M.N. (eds.) Topics in Medical Image Processing and Computational Vision. Lecture Notes in Computational Vision and Biomechanics, vol. 8, pp. 165–180. Springer, Dordrecht/London (2013)
20. Pinho D, Yaginuma T., Lima R.: A microfluidic device for partial cell separation and deformability assessment. BioChip J. **7**, 367–374 (2013)
21. Pires, A.A., Neuhaus, D., Gaehtgens P.: Blood viscosity in tube flow: dependence on diameter and hematocrit. Am. J. Physiol. **263**, H1770–H1778 (1992)
22. Yaginuma T., Oliveira M.S., Lima R., Ishikawa T., Yamaguchi T.: Human red blood cell behavior under homogeneous extensional flow in a hyperbolic-shaped microchannel. Biomicrofluidics **7**(5), 54110 (2013)

Computational Comparison of Algorithms for a Generalization of the Node-Weighted Steiner Tree and Forest Problems

Raul Brás and J. Orestes Cerdeira

Abstract Habitat fragmentation is a serious threat for the sustainability of species. Thus, the identification of effective linkages to connect valuable ecological units is an important issue in conservation biology. The design of effective linkages should take into account that areas which are adequately permeable for some species' dispersal may act as obstructions for other species. The determination of minimum cost effective linkages is a generalization of both node-weighted Steiner tree and node-weighted Steiner forest problems. We compare the performance of different procedures for this problem using large real and simulated instances.

1 Introduction

In conservation biology, habitat fragmentation is considered a key driver of biodiversity loss [4, 10]. To mitigate the impacts of fragmentation on biodiversity, connectivity between otherwise isolated populations should be promoted [15]. To effectively promote connectivity, there is need for procedures to identify linkages (i.e., areas to establish the connection) between habitats of each of several species (i) that take into account that linkage areas for a species might be barriers for others, and (ii) that are cost-efficient, since placing linkage areas under conservation compete with other land uses.

The problem can be formulated as follows. Consider a graph $G = (V, E)$ where the nodes of V identify the cells (usually grid squares) in which the study region has been divided, and which are considered suitable for conservation actions. The edges

R. Brás (✉)
Instituto Superior de Economia e Gestão and Centro de Matemática Aplicada à Previsão e Decisão Económica, Universidade de Lisboa, Portugal
e-mail: rbras@iseg.ulisboa.pt

J.O. Cerdeira
Departamento de Matemática & Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, Portugal
e-mail: jo.cerdeira@fct.unl.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_5

of E define adjacencies between pairs of cells (usually two cells are adjacent if they have a common border).

For each species (or group of "similar" species, i.e., sharing the same habitats and suitable areas to disperse) k, $k = 1, \ldots, m$, let T^k be the set of nodes representing the habitats of species k (*terminals* of type k), and V^k the set of nodes corresponding to cells which can be used as linkage passages for species k. We assume that $T^k \subseteq V^k$, and call k-barriers to the cells of $V \setminus V^k$.

A feasible solution of the problem is a subset of nodes $S \subseteq V$ that, for $k = 1, \ldots, m$, includes a path that only uses nodes of V^k between every pair of nodes in T^k.

Suppose there is a (non negative) cost associated to every node, quantifying the charge of allocating the corresponding cell to conservation purposes. The problem, which we will call multi-type linkage problem (MTLinkP for short), seeks for a minimum cost feasible solution (i.e., which minimizes the sum of the costs of the nodes).

MTLinkP, that was independently considered by Lai et al. [13] and by Alagador et al. [1], is a generalization of the node-weighted Steiner tree [20] and of the node-weighted Steiner forest [8] problems. If $V = V^k$ and $m = 1$ (i.e., only one species, no barriers) MTLinkP is the node-weighted Steiner tree problem. If $V = V^k$, for $k = 1, \cdots, m > 1$ (i.e., different species, no barriers) MTLinkP is the node-weighted Steiner forest problem.

Lai et al. [13] and Alagador et al. [1] proposed a heuristic procedure for MTLinkP by solving a sequence of node-weighted Steiner tree problems, one for each type k, and outcome the union of these m Steiner solutions. We call this approach *type by type*. In the same paper Lai et al. [13] presented a heuristic for MTLinkP that is a generalization of the primal-dual algorithm of Demaine et al. [5] for the node-weighted Steiner forest problem, and report computational results on synthetic instances of small size (up to 15×15 grids) and m up to 5, and a real instance consisting of two species, 4514 cells and up to 17 terminals.

Brás et al. [2] developed a computer application, MulTyLink, implementing a version of the type by type algorithm and a GRASP type heuristic. This software was announced in Brás et al. [3], along with a brief description of the two algorithms used by the program.

In this paper we compare the performances of the primal-dual algorithm of Lai et al. [13] and the two heuristics in MulTyLink on real and simulated data sets. We start giving in Sect. 2 a multiflow formulation for MTLinkP. In Sect. 3 we give some details on the two heuristics of MulTyLink, and summarize the primal-dual heuristic of Lai et al. [13]. In Sect. 4 we report results of computational experiments to compare running times and quality of the solutions produced with the three heuristics. We finish with some remarks in Sect. 5.

2 Multiflow Formulation

Flow formulations have been used to model Steiner tree problems (see, e.g., Wong [21] for the standard edge-weighted Steiner tree problem; Segev [18] for a special case of non negative costs on the edges and negative costs on the nodes and Magnanti and Raghavan [14] for general network design problems with connectivity requirements including the edge-weighted Steiner forest problem).

Here we give a multi-commodity flow based formulation of MTLinkP.

Let $w_v \geq 0$ be a cost associated to each node v of graph G and let $w_v = 0$ if $v \in T^k$, for some $k = 1, \cdots, m$. Denote by A the set of arcs obtained by assigning two arcs of opposite directions to every edge of G.

For $k = 1, 2 \ldots, m$, let $t_1^k, t_2^k, \cdots, t_{p_k}^k$, with $p_k = \left|T^k\right|$, be the terminals of T^k taken by some arbitrary order. Node t_1^k will supply every other node in T^k (the demanding nodes) with one unit of commodity. Variables $f_{(u,v)}^{ki}$ on arcs indicate the amount of commodity k (amount of flow of type k) along arc (u, v) with origin t_1^k and destination t_i^k, $i = 2, \ldots, p_k$. Connectivity of the nodes of T^k in solutions is ensured by the mass balance constraints which state that the amount of commodity k out of a node v minus the commodity k into v must equal the supply/demand amount.

In addition to flow variables $f_{(u,v)}^{ki}$, binary variables x_v on nodes will be used to indicate whether node v is included ($x_v = 1$) or not ($x_v = 0$) in the solution. With these variables MTLinkP can be formulated as follows.

$$\min \sum_{v \in V} w_v x_v \tag{1}$$

subject to:

$$\sum_{\{v \in V^k : (t_1^k, v) \in A\}} f_{(t_1^k, v)}^{ki} = 1, \quad \begin{array}{l} k = 1, \cdots, m, \\ i = 2, \cdots, p_k \end{array} \tag{2}$$

$$\sum_{\{u \in V^k : (u,v) \in A\}} f_{(u,v)}^{ki} = 1, \quad \begin{array}{l} v \in T^k \setminus \{t_1^k\} \\ k = 1, \cdots, m, \\ i = 2, \cdots, p_k \end{array} \tag{3}$$

$$\sum_{\{u \in V^k : (v,u) \in A\}} f_{(v,u)}^{ki} - \sum_{\{u \in V^k : (u,v) \in A\}} f_{(u,v)}^{ki} = 0, \quad \begin{array}{l} v \in V^k \setminus \{T^k\} \\ k = 1, \cdots, m, \\ i = 2, \cdots, p_k \end{array} \tag{4}$$

$$\sum_{\{u \in V^k : (u,v) \in A\}} f_{(u,v)}^{ki} \leq x_v, \quad \begin{array}{l} v \in V^k \setminus \{t_1^k\}, \\ k = 1, \ldots, m, \\ i = 2, \cdots, p_k \end{array} \tag{5}$$

$$x_v \in \{0, 1\}, \qquad v \in V \tag{6}$$

$$f_{(u,v)}^{ki} \geq 0, \quad u, v \in V^k, \ (u, v) \in A, \ k = 1, \ldots, m, \ i = 2, \cdots, p_k. \tag{7}$$

The mass balance equations (2), (3) and (4) dictate that one unit of flow of type k will be routed between the supply node t_1^k and each demanding node of $T^k \setminus \{t_1^k\}$. The "capacity" constraints (5) ensure there is no flow along the arcs entering nodes that are not included in the solution. Constraints (2), (3), (4), (5), (6) and (7) guarantee the existence of a directed path between t_1^k and every other node of T^k, thus ensuring that all nodes of T^k will be in the same connected component of the solution.

We will use the compact formulation (1), (2), (3), (4), (5), (6) and (7) above to derive, from a mixed integer programming solver, MTLinkP optimal values for small size instances, to assess the quality of the solutions produced by the heuristic algorithms of Sect. 3.

3 Heuristics

3.1 Type by Type Heuristic

Given a permutation $i_1, i_2, \ldots, i_m$ of (integer) types $1, 2, \ldots m$, the *type by type* heuristic computes in step k a Steiner solution with respect to $< V^{i_k} >$, the subgraph of G induced by V^{i_k}, updates the costs of nodes v of that solution letting $w_v = 0$, and proceeds to the next step $k + 1$. The final MTLinkP solution results from turning minimal feasible (with respect to inclusion) the union of nodes of the Steiner solutions obtained in each step.

The process can be repeated for a number of different permutations of integers $1, 2, \ldots, m$, and the best solution is returned (see Fig. 1).

To solve the node-weighted Steiner tree problem in each step, we use the following straightforward modification of the well known *distance network heuristic* suggested by Kou et al. [12] for the edge-weighted Steiner tree. If H is a graph

1. $Sol \leftarrow \emptyset$
2. $w(Sol) \leftarrow \infty$
3. $\mathscr{P} \leftarrow$ subset of permutations of $\{1, \ldots, m\}$.
4. For all $P \in \mathscr{P}$
   a. $X \leftarrow \emptyset$
   b. For all $k \in P$
      i. Build graph $< V^k >$ with weights: 0 if $v \in X$, w_v otherwise.
      ii. $X^k \leftarrow$ Steiner solution w.r.t. $< V^k >$ and T^k
      iii. $X \leftarrow X \cup X^k$
   c. Turn X minimal. For each $v \in X \setminus \bigcup_k T^k$ (randomly ordered) remove v from X if $X \setminus \{v\}$ is MTLinkP feasible.
   d. $w(X) \leftarrow \sum_{v \in X} w_v$
   e. If $w(X) < w(Sol)$ then
      i. $Sol \leftarrow X$
      ii. $w(Sol) \leftarrow w(X)$
5. Return Sol.

Fig. 1 TbT heuristic

with costs w on the nodes and terminal set S, we define the distance network $D(S)$ which is the complete graph with S as its node set, and where the weight of every edge (u, v) is the cost of the minimum cost path connecting terminal u to v on H. Note that determining a node-weighted shortest path on undirected graph H between nodes u and v, with $w_u = w_v$, reduces to finding an edge-weighted shortest path from u to v in the digraph obtained assigning opposite directions to every edge of H, and defining the cost of every arc (i, j) as being equal to w_j.

A minimum spanning tree of $D(S)$ is determined and a (node-weighted) Steiner solution N is defined as the set of the nodes of the shortest paths corresponding to the edges of the spanning tree.

In the final step the nodes of N are considered randomly and node j is removed from N if all nodes of S belong to the same connected component of the subgraph of H induced by $N \setminus \{j\}$.

We use the above modification of the *distance network heuristic* since it is fast and does not use large data structures. Procedures such as Klein and Ravi [11] heuristic, based on the Rayward-Smith [16] algorithm, that perform well for node-weighted Steiner problems, would be impractical for the large size instances of MTLinkP we want to handle. Klein and Ravi [11] heuristic needs to compute the minimum cost paths between all pairs of nodes, which is time consuming and requires large amounts of memory.

Lai et al. [13] version of this heuristic uses the Dreyfus-Wagner (DW) algorithm [7] to solve the Steiner problem at each step. DW is an exact dynamic programming algorithm that runs in exponential time, not suitable to solve the instances that we present in this paper.

3.2 Primal Dual Heuristic

Lai et al. [13] gave a modified version of the Demaine et al. [5] heuristic for node-weighted Steiner forest problems. The heuristic operates on the following cut-covering formulation of MTLinkP. Minimize (1) subject to (6), and

$$\sum_{v \in \Gamma^k(S)} x_v \geq f^k(S), \qquad \begin{array}{l} S \subseteq V^k \\ k = 1, \ldots, m \end{array} \tag{8}$$

where $f^k(S) = 1$, if $\emptyset \neq S \cap T^k \neq T^k$ (i.e., if S includes at least one terminal of T^k, but not all), and $f^k(S) = 0$, otherwise, and $\Gamma^k(S)$ is the set of nodes $v \in V^k \setminus S$ adjacent to at least one node in S.

The dual of the linear relaxation of (1), (6), (8) is:

$$\max \sum_{k=1}^{m} \sum_{S \subseteq V^k} f^k(S) y^k(S)$$

1. $X \leftarrow \bigcup_k T^k$
2. For $k = 1, \ldots, m$ calculate $\mathscr{C}(\langle X^k \rangle)$. $y^k(C) \leftarrow 0$ for every $C \in \mathscr{C}(\langle X^k \rangle)$
3. While X is not MTLinkP feasible
   a. Simultaneously increase $y^k(C)$ until, for some v, $\sum_{k=1}^{m} \sum_{C \subseteq V^k : v \in \Gamma^k(C)} y^k(C) = w_v$.
   b. $X \leftarrow X \cup \{v\}$
   c. For $k = 1, \ldots, m$ recalculate X^k and $\mathscr{C}(\langle X^k \rangle)$.
4. Turn X minimal. For each $v \in X \setminus \bigcup_k T^k$ (by reverse order of insertion) remove v from X if $X \setminus \{v\}$ is feasible
5. Return X.

Fig. 2 PD heuristic

subject to:

$$\sum_{k=1}^{m} \sum_{S \subseteq V^k : v \in \Gamma^k(S)} y^k(S) \leq w_v \quad v \in V$$

$$y^k(S) \geq 0 \qquad \begin{array}{l} S \subseteq V^k \\ k = 1, \ldots, m \end{array}$$

The heuristic maintains an infeasible primal solution X, and dual variables $y^k(S)$. The algorithm is described in Fig. 2, where $\mathscr{C}(\langle X^k \rangle)$ are the connected components of the graph induced by $X^k = X \cap V^k$.

3.3 GRASP Heuristic

The type by type (TbT) heuristic and the primal-dual heuristic (PD) of Lai et al. [13] define a feasible solution adding in each step nodes to a current unfeasible solution X. TbT heuristic adds to X a set of nodes that guarantee the connection of all terminals from a certain predetermined type, and assigns costs equal to zero to all the added nodes. PD heuristic adds to X one single node that belongs to V^k and is adjacent to X^k, for at least one not previously determined type k. We present a kind of greedy randomized adaptive search procedure (GRASP) [9] that hybridizes the two heuristics.

GRASP starts with set X consisting of all terminals of T^k, $k = 1 \cdots, m$, and in each step grows the current unfeasible solution X as follows. First, some type k, for which not all terminals of T^k are connected, is uniformly selected. Next, a connected component S of $< X^k >$, the subgraph induced by X^k, that includes terminals of type k, is uniformly selected, and a minimum cost path P, among the minimum cost paths connecting S with every other component of $< X^k >$ that includes terminals of type k, is determined. The nodes of P are added to X, and costs are updated letting $w_v = 0$ to every node v of P. Note that, since the costs of nodes of X are all equal to zero,

1. $w(Sol) \leftarrow \infty$
2. $r \leftarrow$ number of repetitions
3. For $i = 1, \ldots, r$
   a. $X \leftarrow \bigcup_k T^k$
   b. While X is not MTLinkP feasible
      i. $X^k = X \cap V^k$. Calculate $\mathscr{C}(\langle X^k \rangle)$, $k = 1, \ldots, m$
      ii. $Q = \{k :$ not all terminals of T^k belong to the same $S \in \mathscr{C}(\langle X^k \rangle)$, $k = 1, \ldots, m\}$
      iii. If $Q = \emptyset$ then end the while cycle
      iv. $p \leftarrow$ member of Q uniformly selected
      v. $S \leftarrow$ member of $\mathscr{C}(\langle X^p \rangle) : S \cap T^p \neq \emptyset$ uniformly selected
      vi. $P \leftarrow$ minimum cost path connecting S to $U \in \mathscr{C}(\langle X^p \rangle) \setminus S : U \cap T^p \neq / 0$
      vii. $X \leftarrow X \cup P$, $w_v = 0$, $\forall v \in P$
   c. Turn X minimal. For each $v \in X \setminus \bigcup_k T^k$ (randomly ordered) remove v from X if $X \setminus \{v\}$ is MTLinkP feasible.
   d. $w(X) \leftarrow \sum_{v \in X} w_v$
   e. If $w(X) < w(Sol)$ then
      i. $Sol \leftarrow X$
      ii. $w(Sol) \leftarrow w(X)$
4. Return Sol.

Fig. 3 GRASP heuristic

P can be easily obtained with Dijkstra algorithm [6], choosing an arbitrary node in S as the starting node and ending as soon as a node of a component of X^k, including terminals of T^k and different from S, is added to the path.

The final GRASP solution is obtained by turning minimal feasible the solution X obtained in the last step. Given its random behavior, repeating GRASP a number of times with the same input is likely to produce different solutions, and the best solution is outcome (see Fig. 3).

4 Computational Experiments

We performed computational tests to evaluate the quality of the solutions produced by the heuristics, as well as the practicality of the flow formulation of Sect. 2.

4.1 General Case

Here we report results for the case where not all V^k coincide.

4.1.1 Data

We used real and simulated instances to test the heuristics.

1. Real Data.

Real data concerns the linkage of climatically-similar protected areas (PA) in the Iberian Peninsula (IP). IP is represented as 580,696 1 km × 1 km cells, from which 80,871 cells intersecting the 681 existent PA in the IP were defined as terminals. Terminals were clustered in four groups sharing similar climates (with respect to four climatic variables which are considered important drivers of species' distributions). Adjacency was considered in terms of common edges or corners of the square cells.

Cells with considerable human intervention (values derived from Human Footprint Index data available from http://www.ciesin.columbia.edu/wild_areas greater than 60 in a range from 0 to 100) were excluded as they were considered poorly permeable to species' movements. This has reduced the number of cells to 438,948 (which includes every protected cell).

Figure 4 (page 74) shows the location of protected cells from each class (colored cells), and cells that were excluded due to presenting high levels of human intervention (grey cells).

For $k = 1, \ldots, 4$, V^k was defined as the set of cells that do not significantly differ from the mean climatic conditions of PA of class k. This was delineated as follows. The centroid, in the climatic space, of the PA cells of each climatic class was defined, and the Euclidean distances from the climate conditions of each cell

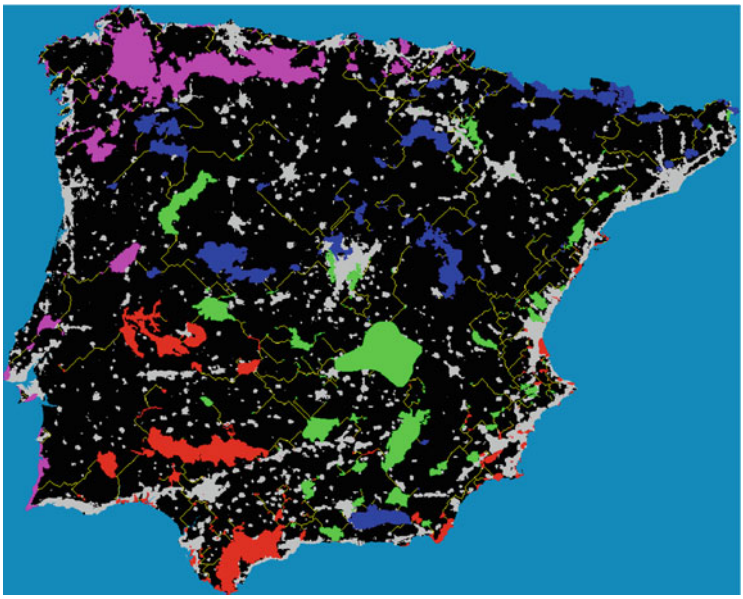


Fig. 4 Iberian Peninsula data (scenario 2). Protected cells are colored *red*, *green*, *blue* and *magenta*. *Grey* areas represent cells not in V. Cells of the solution obtained with TbT heuristic are colored *yellow*

to the centroid of each class were computed. This retrieved four values $d^k(v)$, for each cell v, expressing the dissimilarity of cell v to every climatic class k. Cell $v \in V^k$ (i.e., v was not considered k-barrier) if $d^k(v)$ is below a certain threshold value B^k. Two scenarios were considered. In scenario 1, B^k was defined as the largest dissimilarity $d^k(v)$, among the protected cells v in every PA from class k. In scenario 2, B^k was set as the third quartile of the $d^k(v)$ values for protected cells v of class k. Cell u was included in V^k (i.e., u was not considered k-barrier) if u belongs to some PA of class k, or $d^k(u) < B^k$.

The rational for the identification of linkages between climatically-similar protected areas, free from climatic barriers, stands on the assumption advocated by Alagador et al. [1] that species with similar ecological requirements occupy the same environments. Thus, linking climatically-similar protected areas is an effective way to promote the dispersal of species, counteracting in part the negative effects of fragmentation.

A cost was assigned to every non protected cell that is proportional to the cell's fraction not covered by Natura 2000 Network (w_v = (100−percentage of Natura 2000 Network covered by v)/100). The Natura 2000 network is a European scaled conservation scheme designed to complement nationally-defined protected areas. We assigned cost equal to zero to every protected cell.

Details on the IP data can be found in Alagador et al. [1].

We denote by $IP1$ and $IP2$ the IP instances under scenarios 1 and 2, respectively.

2. Simulated Data.

Simulated data were generated as follows. Each node of V is a cell from a $n \times n$ grid. Two cells are adjacent if they have a common edge or corner.

To define V_k we start by uniformly selecting an integer $s \in [0, m]$ and assume that species $1, \ldots, s$ are "specialist" (can only thrive in a narrow range of environmental conditions) and species $s + 1, \ldots, m$ are "generalist" (are able to thrive in a wide variety of environmental conditions). Each node v of V is included in V_k with probability $1/4$ for each "specialist" species $k \leq s$, and with probability $3/4$ for each "generalist" species $k > s$. The number of terminals of each type was obtained from a discrete uniform distribution in interval $[2, \max\{|V|/1000, 5\}]$, and terminals chosen uniformly among the nodes of V^k.

We assigned to every node in $V \setminus (\cup_k T^k)$ a cost from an uniform distribution in $[0, 1]$, and cost zero to every node of T^k.

We generated small instances with $n = 10, 20, 30, 40, 50$ and $m = 2, 3, 4, 6, 8, 10$ and large instances with $n = 100, 200, 300, 400, 500$ and $m = 4, 6, 8, 10$.

For the same values of n and m we generated 10 instances. This gave a total of 500 instances.

4.1.2 Results

Here we report the main results of the computational tests that we carried out.

Heuristics were implemented in C++, using the Boost Graph Library [19] to calculate spanning trees, shortest paths and connected components. Parallel programming was not used and so they ran in a single thread. All times refer to elapsed times. The computers were dedicated to running the instances, so that elapsed time is close to CPU time. Solutions for the Iberian Peninsula data were obtained with a Intel Core2 Quad CPU Q9450 @2.66 GHz and 4 GB of memory machine, while for simulated data the solutions were obtained in a machine with 2 AMD Opteron 6172 processors (24 cores) @2.1 GHz and 64 GB of memory.

1. Real Data.

Table 1 displays results obtained for the Iberian Peninsula's data with each of the three heuristics. The first column identifies the problem instance (scenarios 1 and 2). Each of the three pairs of the remaining columns contains the value of the solution obtained with a heuristic: GRASP, type by type (TbT) and primal-dual (PD), and the corresponding running time in seconds. The TbT heuristic ran for every permutation of the $m = 4$ types, while GRASP was limited to 2 hours of execution. We let the program finish the current repetition i, if it has started before the time expired, thus computation times can exceed 7200 seconds. PD was not time-limited in order to produce a solution.

GRASP obtained the best solutions. The costs of the solutions produced by TbT were slightly higher, but the times to run the 24 permutations of the four types were lower than the 2 hours that limited the execution of GRASP. PD had a poor performance. Long computation times were necessary to obtain solutions with costs that are greater than those of the solutions obtained with GRASP and with TbT. This negative behavior of PD can be explained by the specific structure of these graphs. Nodes which are far apart on the grid are connected by long paths. Thus, it is likely that PD includes a large number of redundant nodes until a feasible solution is reached. Solutions with many redundant nodes are difficult to turn minimal. The process is time consuming and produces poor solutions. GRASP and TbT, in each step, add to the solution that is being constructed the nodes of a minimum cost path connecting a pair of terminals. Thus, the number of redundant nodes is likely to be much less than that generated by PD.

Figure 4 shows a solution, obtained with the TbT heuristic, for the IP2 instance. Protected cells are colored red, green, blue and magenta and the cells of the solution

Table 1 Results for the Iberian Peninsula

	GRASP		TbT		PD	
Instance	Cost	Time	Cost	Time	Cost	Time
*IP*1	2024.67	7782.55	2035.73	5012.39	2162.11	544,490.00
*IP*2	2121.03	7525.25	2148.49	7075.90	2167.62	347,003.00

are yellow. Grey areas represent cells not in V (human footprint over 60). For a detailed interpretation of the solution, knowledge of the location of the barriers from each type would be needed.

2. Simulated Data.

The main results derived with small and large instances for simulated data are given in Tables 2 and 3, respectively. Recall that 10 instances with the same values of $|V|$ and m were considered and, therefore, each row of Tables 2 and 3 summarizes the results of 10 instances.

Table 2 Results for small instances

$\|V\|$	m	#Opts	GRASP			TbT				PD			
			% dev. from			% dev. from				% dev. from			
			Opt	BestH	Best	Opt	BestH	Best	Time	Opt	BestH	Best	Time
100	2	10	1.67	1.67	9	1.67	1.67	9	0.00	2.15	2.15	9	0.00
	3	10	1.66	1.66	9	1.66	1.66	9	0.01	0.00	0.00	10	0.00
	4	10	4.02	1.94	8	3.87	1.79	9	0.06	6.49	4.20	6	0.00
	6	10	2.32	2.17	9	3.31	3.16	8	5.19	1.61	1.48	7	0.00
	8	10	1.21	0.00	10	1.21	0.00	10	44.43	2.06	0.86	5	0.01
	10	10	0.31	0.31	9	1.27	1.27	7	60.51	0.75	0.75	8	0.01
400	2	10	0.34	0.00	10	2.51	2.18	5	0.01	7.53	7.17	5	0.02
	3	10	1.80	0.25	9	4.55	2.90	5	0.01	11.47	9.71	4	0.02
	4	10	1.60	0.00	10	4.48	2.79	5	0.06	9.02	7.27	3	0.03
	6	10	2.08	0.95	8	3.56	2.38	7	1.49	9.06	7.87	1	0.04
	8	10	0.90	0.10	9	3.70	2.87	6	18.21	9.50	8.62	2	0.06
	10	10	2.24	0.00	10	6.33	3.95	3	43.46	11.20	8.69	0	0.15
900	2	10	0.85	0.00	10	2.60	1.72	6	0.01	5.95	4.96	6	0.07
	3	10	0.54	0.00	10	1.82	1.28	8	0.02	8.15	7.50	5	0.09
	4	10	1.34	0.00	10	4.66	3.26	4	0.08	15.41	13.82	3	0.14
	6	8	2.71	0.19	9	6.54	4.55	3	3.57	19.06	15.32	2	0.27
	8	7	2.86	0.00	10	4.70	2.05	2	14.31	13.09	13.72	0	0.37
	10	7	1.82	0.39	8	3.25	2.75	5	42.45	10.26	11.07	2	0.56
1600	2	7	1.49	0.22	9	1.92	1.34	4	0.02	7.93	7.66	3	0.22
	3	5	0.09	0.00	10	0.49	4.01	4	0.05	0.65	8.51	4	0.36
	4	6	0.69	0.20	8	1.17	2.38	5	0.21	5.92	10.07	5	0.53
	6	5	1.52	0.43	8	1.93	1.13	5	4.33	4.39	16.84	3	0.74
	8	5	0.29	0.00	10	1.49	2.80	3	12.44	10.56	11.81	2	0.77
	10	1	0.00	0.06	9	1.47	4.27	1	36.13	3.32	15.09	0	1.96
2500	2	4	0.77	0.00	10	0.77	3.35	5	0.04	0.77	5.62	4	0.75
	3	6	0.00	0.08	9	0.98	1.11	6	0.07	0.98	6.91	4	0.67
	4	6	1.52	0.51	9	0.64	2.39	6	0.24	1.96	6.48	4	0.78
	6	4	5.05	0.15	8	5.57	2.41	3	6.01	4.65	7.39	4	1.71
	8	2	1.24	0.00	10	3.35	3.83	2	17.60	22.30	15.73	2	2.60
	10	0		0.04	8		1.59	6	16.20		16.19	3	2.60

Table 3 Results for large instances

$\|V\|$	m	GRASP		TbT			PD		
		%dev.	Best	%dev.	Best	Time	%dev.	Best	Time
10,000	4	0.00	10	5.27	1	1.56	18.17	1	43.71
	6	0.00	10	4.55	1	49.93	19.91	1	89.57
	8	0.00	10	3.43	2	214.82	11.90	2	67.37
	10	0.00	10	4.86	0	766.98	21.31	0	148.57
40,000	4	0.00	10	3.13	2	52.95	14.74	2	1292.83
	6	0.00	10	3.50	0	732.16	7.55	0	1116.29
	8	0.00	10	3.44	1	982.21	14.08	1	1709.82
	10	0.00	10	3.59	1	1320.94	13.60	1	1998.23
90,000	4	0.00	10	2.46	3	185.65			
	6	0.00	10	2.21	2	1056.98			
	8	0.00	10	3.66	0	1654.99			
	10	0.13	9	3.73	1	1823.81			
160,000	4	0.00	10	2.24	4	819.54			
	6	0.00	10	2.42	2	1325.13			
	8	0.03	9	2.34	2	1931.62			
	10	0.00	10	2.14	0	1947.12			
250,000	4	0.00	10	3.26	0	1378.50			
	6	0.00	10	2.45	0	1821.26			
	8	0.00	10	2.76	0	2053.90			
	10	0.00	10	2.09	2	2297.54			

We established common elapsed time limit values for the heuristics. One minute for small instances and 30 minutes for large instances, but we allowed GRASP to finish the current repetition i, and TbT to finish the current permutation P.

For most of small instances we were able to obtain optimal solutions from the flow formulation (1), (2), (3), (4), (5), (6) and (7), using CPLEX 12.4, with parallel mode set to opportunistic and 24 parallel threads (all other options used default values). For each instance, CPLEX execution time-limit was set to 1 hour of elapsed time, meaning up to 24 hours of CPU time since the machine has 24 cores.

In Table 2 column *#Opts* indicates the number of instances for which optimal solutions were found. The two columns labeled *% dev. from* indicate, for each heuristic, the mean of the relative deviations (in percentage) from the optimal values (*opt*) and from the best values of the heuristic solutions (*bestH*). The relative deviation is calculated by the expression $100(h - w^*)/w^*$, where h is the value of the heuristic solution, and w^* is the optimal value (*opt*), or the minimum of the values of the three heuristic solutions (*bestH*), respectively. The number of optimal values with respect to which averages were computed is given in column *#Opts*. Columns *best* report the number of instances for which the heuristic found the best value among the values of the three solutions obtained for the same instance with the three heuristics. Columns *time* indicate the mean computation times (in seconds) for TbT and PD. The computation times are not reported for GRASP since it uses all the amount of time allowed.

Table 3 is similar except that there are no columns regarding optimal values, since CPLEX was unable to handle the large instances. Thus, columns *% dev.* and *best* refer to comparisons with the best values of heuristic solutions.

GRASP was clearly superior for the instances considered, while PD had a poor performance.

For small instances the average over the 30 values of column *% dev. from opt* in Table 2 was 1.48 for GRASP, 2.81 for TbT and 7.11 for PD. Only four of these 30 values exceeded 2.5 % for GRASP, while six values exceeded 4.5 % for TbT. GRASP was a best heuristic in at least eight instances out of the 10 with the same $|V|$ and m. Considering all the 300 small instances, GRASP was a best heuristic in 275, TbT in 161 and PD in 116.

For the large instances the superiority of GRASP was even more evident. It has obtained the best results in 198 out of 200 instances. The mean relative deviations between TbT results and the best heuristic values were always below 5.3 %, but it attained the best result only on 24 instances.

Results on simulated data confirmed the bad behavior of PD with this kind of instances. For $|V| \geq 90{,}000$, we were unable to find solutions within the time limit of 30 minutes, except for a few instances. These were not considered in order to not bias the analysis of the results. The corresponding entries are blank on Table 3. In general, solutions were of poor quality. It seems that PD has difficulties dealing with instances where graphs have the structures here considered. An explanation was previously given when analyzing the results on the IP instances.

A fact that should be mentioned is that, several times, TbT succeeded to complete its computations within the time limits established, despite the relative high values of m ($m = 8, 10$). This is justified by the way instances were generated. Each cell of the $n \times n$ grid belongs to V^k with probability 1/4 for "specialized" species k and 3/4 for "generalist" species k. Thus, it may happen that all components of the subgraph induced by V^k, particularly for "specialized" species k, include at most one terminal of T^k, i.e., every path connecting any two terminals of T^k include some k-barrier. In this case there is no need to consider species k, as no two terminals of T^k can be linked in V^k. Since "specialist" species were uniformly chosen among the m species, the number of species that needs to be considered might be much smaller than m.

4.2 Case Where All V^k Coincide

MTLinkP is a generalization of the node-weighted Steiner tree and of the node-weighted Steiner forest problems. Therefore, GRASP, TbT and PD can be used, with no modification, to solve those problems.

We carried out some computational tests to assess how the heuristics perform on solving node-weighted Steiner forest problem.

For node-weighted forest problem, the PD heuristic is the Demaine et al. [5] algorithm. Another heuristic, based on the Rayward-Smith algorithm [16, 17] that performs well in practice for node-weighted Steiner forest is the Klein and Ravi [11] heuristic.

Klein and Ravi heuristic (KR) begins by computing the matrix M of the costs of minimum cost paths between every pair of nodes in V. Then, starting with X consisting of all terminals of T^k, $k = 1, \cdots, m$, in each step, KR adds to X the nodes of certain paths that connect a number of connected components of $< X >$, the subgraph induced by X. The connected components to merge are selected from the values of a function f that is calculated as follows, for every node $v \in V$. Let $\mathscr{S}$ be the set components of $< X >$ that, for some k, includes at least one node of T^k but not all, and let $\mathscr{S}_r$ be the family of all r sets of $\mathscr{S}$ (i.e., if $S_r \in \mathscr{S}_r$, $|S_r| = r$). For every $v \in V$ and $S_r \in \mathscr{S}_r$, let $w(v, S_r)$ be the sum of the costs of minimum cost paths connecting v with each of the r components in S_r. For every $v \in V$, define $f(v, r) = \min_{S_r} w(v, S_r) - (r-1)w_v$. The value $f(v, r)$ is the minimum cost of merging r components of $\mathscr{S}$ with r paths rooted at v. Note that the computation of $f(v, r)$ can be quickly achieved from matrix M. Finally, $f(v) = \min_{2 \le r \le |\mathscr{S}|} f(v, r)/r$, which is the minimum of the mean values of $f(v, r)$ with respect to r. In each step, KR adds to X the nodes of the paths from v which minimizes $f(v)$, while $\mathscr{S}$ is not the empty set. When there are no more components to merge, the heuristic proceeds turning solution X minimal.

We compared the performances of GRASP, TbT, Demaine et al. [5] (PD) and Klein and Ravi [11] (KR) heuristics on instances generated as above for simulated data, except that $V^k = V$, for $k = 1, \ldots, m$. We considered $n \times n$ grid graphs with $n = 50, 100, 200$ and $m = 2, 4, 6, 8, 10$ types of terminals. For each n and m two instances were generated. Table 4 reports costs and times (in seconds) on each instance. GRASP and TbT heuristics were restricted to 30 minutes of execution time. Computations were processed with the same machine that was used for the simulated data.

Results for GRASP and KR were very similar. KR obtained the best result in 56.7 % of the cases, while GRASP was the best heuristic in 40.0 % of the cases and PD in one case (3.3 %). TbT never obtained the best result. The mean relative gap between the value vH obtained by the heuristic H and the value vKR obtained with KR, given by $(vH - vKR)/vKR$, was 0.5 % for H = GRASP, 4.4 % for H = TbT and 3.5 % for H = PD. Considering only the cases for which KR performed better than heuristic H ($vKR < vH$), the mean of the relative gap was 3.1 % for H = GRASP, 5.2 % for H = TbT and 3.9 % for H = PD.

Results showed that GRASP performed better than KR in smaller instances, while in general KR outperforms GRASP for larger ones. However, the relative gap did not exceed 8.1 % (for an instance where $n = 200$ and $m = 8$). The values obtained by TbT were slightly greater than those produced by GRASP. This was more evident for larger instances ($n = 200$). PD obtains good results in the larger instances. For $n = 200$ and $m \ge 4$ it obtains better results than GRASP with relatively small times of execution. KR heuristic maintains in memory matrix M of the costs of minimum cost paths between every pair of nodes in V. For $n = 200$

Table 4 Results for Steiner forest

$\|V\|$	m	GRASP		TbT		PD		KR	
		Cost	Time	Cost	Time	Cost	Time	Cost	Time
2500	2	19.32	1800.28	20.12	0.11	20.09	1.15	19.20	2.04
		12.61	1800.37	13.01	0.08	15.92	2.34	14.67	1.92
	4	24.45	1800.18	25.31	2.64	26.36	1.57	25.04	2.68
		22.78	1800.48	23.23	2.46	25.53	1.87	22.98	2.33
	6	24.39	1800.65	24.83	104.46	25.38	0.82	24.64	2.73
		30.03	1800.47	30.74	98.04	31.56	1.47	30.44	2.85
	8	36.56	1801.00	38.66	1800.60	38.32	1.67	36.86	4.58
		29.65	1800.23	30.27	1800.37	30.50	0.81	29.97	3.57
	10	34.30	1800.63	37.12	1801.13	36.90	1.82	35.51	3.82
		33.46	1801.23	34.80	1800.23	33.51	1.01	33.30	4.71
10,000	2	46.64	1800.54	48.39	0.73	48.02	17.43	46.83	39.35
		42.19	1800.91	43.96	0.78	43.16	25.60	41.97	48.66
	4	62.06	1800.88	65.60	18.54	65.36	26.49	61.77	58.09
		47.44	1801.16	50.20	12.94	49.05	22.79	47.35	48.61
	6	78.49	1800.33	79.95	956.15	81.00	42.66	77.16	64.57
		67.07	1801.34	68.97	655.02	68.02	18.02	67.31	60.66
	8	74.07	1800.78	75.25	1801.68	75.34	17.31	71.36	72.82
		79.73	1800.39	81.12	1801.94	82.17	22.02	78.56	82.25
	10	100.02	1801.10	100.70	1802.77	99.83	59.72	98.07	96.04
		90.77	1801.33	95.66	1802.23	92.31	32.39	89.00	96.32
40,000	2	178.80	1801.51	192.27	9.94	189.42	501.73	189.39	1541.14
		83.15	1801.27	87.51	3.47	89.15	328.08	84.28	1154.73
	4	257.27	1806.26	265.71	190.69	253.92	1155.15	273.20	2280.98
		188.07	1800.93	197.61	151.75	187.33	349.34	182.06	1666.43
	6	337.71	1813.55	348.49	1814.00	327.30	882.97	317.66	4884.35
		225.93	1803.42	236.66	1806.29	228.74	829.09	218.27	2187.45
	8	316.43	1815.54	327.89	1821.25	303.15	686.42	292.69	4239.43
		305.71	1802.64	322.56	1812.95	300.84	515.96	291.56	3982.47
	10	297.26	1813.47	310.58	1813.78	290.70	885.06	283.34	3827.51
		372.83	1804.53	386.94	1817.54	360.27	597.45	345.79	6483.80

6 GB of memory are needed, and for $n = 300$ 30 GB are needed. Thus, KR heuristic could not be used for the real IP instances.

Given the above limitations of KR, GRASP and PD appear to be good options to solve large node-weighted Steiner forest problems for the type of graphs here considered.

5 Conclusions

In this paper we considered a mixed integer flow formulation and three heuristics for MTLinkP. The flow based formulation only permited to solve instances up to 2500 nodes, which is far below the size of the instances that occur in the context of conservation biology. For the specific structure of graphs of the instances that occur in conservation, GRASP seems to be a good option. Producing different solutions from different runs, on reasonable times, is relevant since, rather than a single solution, decision making needs to consider different options before proceeding negotiations with stakeholders. There are many issues (e.g., socioeconomic) involved in the analysis of conservation actions which are not easily quantifiable, thus having different alternatives to choose is an important feature.

Acknowledgements We are grateful to Maria João Martins and Diogo Alagador for discussion and assistance. Both authors were supported by the Portuguese Foundation for Science and Technology (FCT). R. Brás was funded by the project PEst-OE/EGE/UI0491/2013 and the *CEMAPRE (Centro de Matemática Aplicada à Previsão e Decisão Económica)* under the FEDER/POCI Programme. J. O. Cerdeira was funded through the projects UID/MAT/00297/2013, *CMA (Centro de Matemática e Aplicações)* and PTDC/AAC-AMB/113394/2009.

References

1. Alagador, D., Triviño, M., Cerdeira, J.O., Brás, R., Cabeza, M., Araújo, M.B.: Linking like with like: optimizing connectivity between environmentally-similar habitats. Landsc. Ecol. **27**(2), 291–301 (2012)
2. Brás, R., Cerdeira, J.O., Alagador, D., Araújo, M.B.: Multylink, version 2.0.2 (2012). (computer software http://purl.oclc.org/multylink)
3. Brás, R., Cerdeira, J.O., Alagador, D., Araújo, M.B.: Linking habitats for multiple species. Env. Model. Softw. **40**, 336–339 (2013)
4. Brooks, T.M., Mittermeier, R.A., Mittermeier, C.G., Rylands, A.B., da Fonseca, G.A.B., Konstant, W.R., Flick, P., Pilgrim, J., Oldfield, S., Magin, G., Hilton-Taylor, C.: Habitat loss and extinction in the hotspots of biodiversity. Conserv. Biol. **16**, 909–923 (2002)
5. Demaine, E., Hajiaghayi, M., Klein, P.: Node-weighted steiner tree and group steiner tree in planar graphs. In: Albers, S., et al. (eds.) Automata, Languages and Programming. Lecture Notes in Computer Science, vol. 5555, pp. 328–340. Springer, Berlin/Heidelberg (2009)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
7. Dreyfus, S.E., Wagner, R.A.: The Steiner problem in graphs. Networks **1**(3), 195–207 (1971)
8. Duin, C.W., Volgenant, A.: Some generalizations of the Steiner problem in graphs. Networks **17**(3), 353–364 (1987)
9. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. J. Glob. Optim. **6**(2), 109–133 (1995)
10. Hanski, I.: The Shrinking World: Ecological Consequences of Habitat Loss. Excellence in Ecology, vol. 14. International Ecology Institute, Oldendorf/Luhe (2005)
11. Klein, P., Ravi, R.: A nearly best-possible approximation algorithm for node-weighted Steiner trees. J. Algorithms **19**(1), 104–115 (1995)

12. Kou, L., Markowsky, G., Berman, L.: A fast algorithm for Steiner trees. Acta Inf. **15**, 141–145 (1981)
13. Lai, K.J., Gomes, C.P., Schwartz, M.K., McKelvey, K.S., Calkin, D.E., Montgomery, C.A.: The Steiner multigraph problem: wildlife corridor design for multiple species. In: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, vol. 2, pp. 1357–1364 (2011)
14. Magnanti, T.L., Raghavan, S.: Strong formulations for network design problems with connectivity requirements. Networks **45**(2), 61–79 (2005)
15. Merriam, G.: Connectivity: a fundamental ecological characteristic of landscape pattern. In: Brandt, J., Agger, P. (eds.) Proceedings of the 1st International Seminar on Methodology in Landscape Ecological Research and Planning, pp. 5–15. Roskilde University, Denmark (1984)
16. Rayward-Smith, V.J.: The computation of nearly minimal Steiner trees in graphs. Int. J. Math. Educ. Sci. Technol. **14**(1), 15–23 (1983)
17. Rayward-Smith, V.J., Clare, A.: On finding Steiner vertices. Networks **16**(3), 283–294 (1986)
18. Segev, A.: The node-weighted steiner tree problem. Networks **17**, 1–17 (1987)
19. Siek, J.G., Lee, L., Lumsdaine, A.: The Boost Graph Library: User Guide and Reference Manual. Addison-Wesley Longman, Boston (2002)
20. Winter, P.: Steiner problem in networks: a survey. Networks **17**, 129–167 (1987)
21. Wong, R.: A dual ascent approach for Steiner tree problems on a directed graph. Math. Progr. **28**, 271–287 (1984)

Development of a Numerically Efficient Biodiesel Decanter Simulator

Ana S.R. Brásio, Andrey Romanenko, and Natércia C.P. Fernandes

Abstract This chapter deals with the modelling, simulation, and control of a separator unit used in the biodiesel industry. While mechanistic modelling provides an accurate way to describe the system dynamics, it is an iterative and computationally burdensome process that arises from the need to determine the liquid-liquid equilibria via the *flash* calculation. These disadvantages would preclude the use of mechanistic models for process optimization or model based control. In order to overcome this problem, an alternative strategy is here suggested. It consists of maintaining the mechanistic model structure and to approximate the iterative calculations with an artificial neural network. The general approach for dataset consideration and neural network training and validation are presented. The quality of the resulting neural network is demonstrated to be high while the computation burden is significantly reduced. Finally, the obtained grey-box model is used in order to carry out dynamic simulation and control tests of the unit.

1 Introduction

Today's society is largely dependent on oil as an energy source. However, since it is a finite resource, renewable alternatives have been playing an increasingly important role. In particular, biodiesel—a biofuel produced from oil—presents a set of very attractive characteristics. In this context, a considerable research effort devoted to the development and well-functioning of biodiesel industry has been carried out in recent years.

A.S.R. Brásio
CIEPQPF, Department of Chemical Engineering, University of Coimbra, Portugal & Ciengis, Coimbra, Portugal
e-mail: ana.brasio@ciengis.com

A. Romanenko
Ciengis, Coimbra, Portugal
e-mail: andrey.romanenko@ciengis.com

N.C.P. Fernandes (✉)
CIEPQPF, Department of Chemical Engineering, University of Coimbra, Coimbra, Portugal
e-mail: natercia@eq.uc.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_6

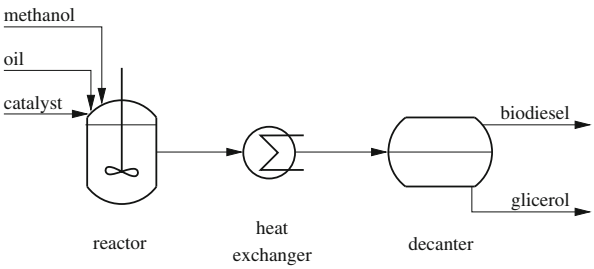


Fig. 1 Simplified schematic representation of the batch biodiesel production process

One of the most relevant units of a biodiesel production line is the reactor, where the oil reacts with methanol under certain operating conditions to produce a mixture of biodiesel and the byproduct glycerol. After the reaction, the mixture is cooled down and its components separated. Figure 1 represents schematically the production line. It should be noted that the separation step in biodiesel industry is commonly performed in a gravity settler. The gravitational settling is a lengthy process and therefore this step represents a significant part of the total production time, exceeding several times the residence time required in the reactor.

A decrease in settling time would represent an economic process improvement. With this motivation, it is appealing to use dynamic optimization tools [3] in order to reach a compromise between the objectives sought and the costs associated with them. These techniques are based on models that describe the dynamics of the process and that can achieve a high degree of complexity. Also, the operation of a biodiesel production line can be greatly improved by a system of non-linear predictive control as described in [6]. Such system requires a set of dynamic models of existing process units in the production line. These models are used to obtain predictions of the temporal trajectories of the state variables and of the output variables as well as to determine the sensitivity of the solutions relatively to the initial state and to the manipulated variables.

In the decanter two liquid phases coexist (the light and the heavy phases) which interact with each other. It is therefore necessary to carry out calculations of liquid-liquid equilibrium in order to quantify this interaction for subsequent incorporation of this phenomenon in the dynamic model of the decanter. Quantification of liquid-liquid equilibria is carried out by the *flash* calculation [11], which is an iterative method.

However, a dynamic model which employs iterative methods cannot be integrated efficiently in a predictive control computing platform. The model is invoked dozens of times per iteration. Although the system has mechanisms to accelerate the convergence of the algorithm, the use of iterative tools significantly increases the calculation time and the required memory. On one hand, the iterative calculation of phase equilibrium on each invocation of the model precludes its use from the standpoint of required computation times. On the other hand, the iterative form worsens the use of automatic differentiation methods, as ADOL-C [14] or CppAD [2], because it significantly increases the memory needed to perform the calculations.

An alternative approach to the calculation of phase equilibria in order to avoid the iterative method without deteriorating the quality of predictions is presented here. The results obtained by *flash* calculations are approximated by a model based on neural networks whose type, composition and characteristics are detailed in Sect. 3. Finally, a first-principle dynamic model of a decanter is implemented applying the developed network to characterize the phase equilibrium. Such model is then used to study the decanter dynamics by simulation.

2 Liquid-Liquid Equilibrium

The methodology most commonly used to quantify the liquid-liquid equilibrium is the *flash* calculation described in detail in [11] for situations of equilibrium between two partially miscible liquid phases.

Considering a feed flow containing n_c components with composition $x_{i,\text{in}}$, the equilibrium at pressure P and temperature T is reached forming two distinct phases (light and heavy) with composition $x_{i,\text{lt}}$ and $x_{i,\text{hv}}$, respectively (where $i = 1, \ldots, n_c$). The feed is separated into two phases: the molar fraction L_{lt} constitutes the light phase and the remaining fraction $1 - L_{\text{lt}}$, is the heavy phase. The equilibrium of each component in the mixture is set by K_i which represents the ratio of the molar fractions of chemical species i in the two liquid phases, i.e.

$$K_i = \frac{x_{i,\text{lt}}}{x_{i,\text{hv}}} = \frac{\gamma_{i,\text{lt}}}{\gamma_{i,\text{hv}}}\,, \tag{1}$$

where $\gamma_{i,\text{lt}}$ and $\gamma_{i,\text{hv}}$ are the activity coefficients of component i in the light and heavy phases, respectively.

Figure 2 represents schematically the mechanistic quantification of liquid-liquid equilibrium, giving evidence of its iterative nature. After specification of the feed, and already inside the iterative cycle, the UNIFAC method (or one of its variations) is used to calculate the activity coefficients required to the calculation of the equilibrium constants. The UNIFAC method [9] estimates the coefficients based on the sum of the contributions of functional groups present in the mixture components: ester (biodiesel), methanol and glycerol.

The oil that origins the ester is composed of glycerides (mainly triglycerides) whose skeleton consists of a glycerol molecule binding fatty acids. The oil, having a biological origin, is characterized by natural variability. However, the lauric acid is normally the fatty acid in greater quantities in vegetable oils. For this reason and in the context of this study, it is considered that the fatty acid present in the raw material is only lauric acid (i.e., biodiesel consists exclusively of ester methyl laurate).

Once convergence for the *flash* calculation is reached, it is then possible to quantify the degree of separation of component i by the light and the heavy phases.

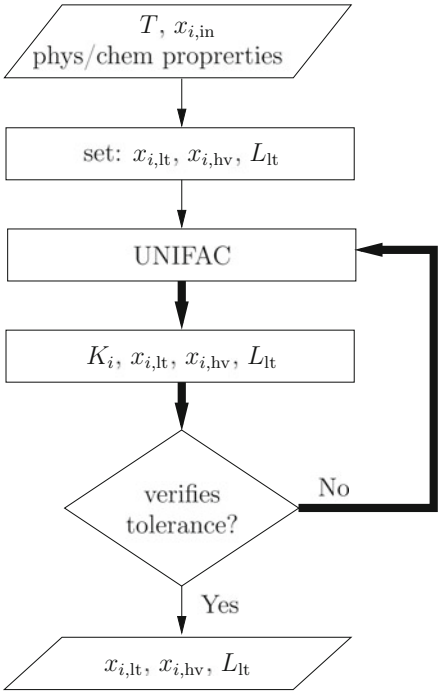


Fig. 2 Flowchart of the *flash* method to determine the liquid-liquid equilibrium

From the amount initially present, the fraction of component *i* that goes to the light phase is given by

$$\xi_i = L_{\text{lt}} \frac{x_{i,\text{lt}}}{x_{i,\text{in}}} \,. \tag{2}$$

3 Artificial Neural Network

Artificial neural networks are used in many engineering applications for predicting variables of complex systems [7, 8]. Their use allows the simulation of physical phenomena without explicit mechanistic formulation to describe the relationships between the variables [8].

Feedforward back-propagation are the neural networks most used in these applications. Such networks are considered static because they depend only on the current input variables and constants. The absence of further information (*feedback*) ensures the stability of the model [7].

Figure 3 represents the architecture of the neural network of the type *feedforward back-propagation* here implemented. It consists of three layers of neurons or nodes (the input layer, the intermediate layer and the output layer), as is typical of this type of networks.

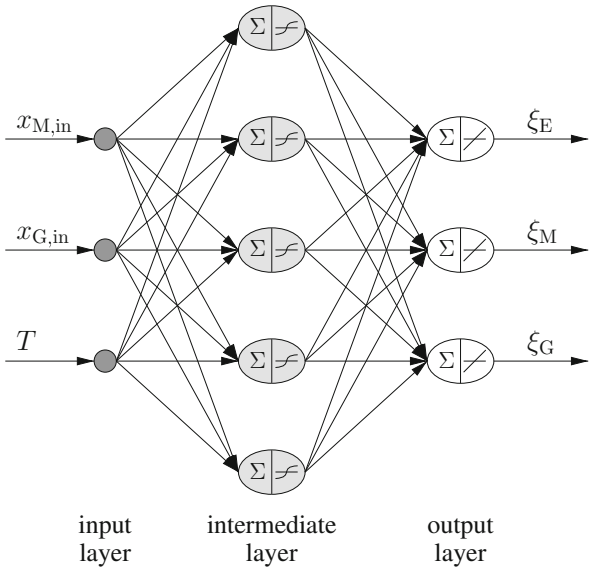


Fig. 3 Neural network used to substitute the *flash* calculation

The number of intermediate layers can vary increasing the prediction capacity of the network, which proves particularly useful in problems with a large number of input variables. However, an increase in the number of these layers also contributes to the over-training of the network due to the large number of constants to determine, that is, it can lead to the *overfitting* of the network [4, 7], apart from increasing exponentially its learning time.

Intermediate and output neurons are structured by an aggregate function and an activation function. According to standard practice, the sum is used in the present work as aggregation function. In what respects the activation function, the most common are the linear, the sigmoid and the hyperbolic tangent functions [7]. Here, the hyperbolic tangent and the linear function were chosen for intermediate and output layers, respectively, as shown within the referred neuron in Fig. 3.

Each neuron is directly connected to the neurons of adjacent layers. Each link is assigned a weight that represents the degree of relationship between the two neurons involved. Mathematically, one can write [4, 7]

$$I_j = f_{\mathrm{I}}\left(\sum_{i=1}^{n_{\mathrm{X}}} w_{ij}\cdot X_i + \theta_j\right), \quad j = 1,\ldots,n_{\mathrm{I}} \tag{3}$$

and

$$Y_k = f_{\mathrm{Y}}\left(\sum_{j=1}^{n_{\mathrm{I}}} W_{jk}\cdot I_j + \Gamma_k\right), \quad k = 1,\ldots,n_{\mathrm{Y}}\,, \tag{4}$$

where n_{X} is the number of input neurons, n_{I} is the number of intermediate neurons, n_{Y} is the number of output neurons, X_i is the input neuron i, I_j is the intermediate neuron j, Y_k is the output neuron k, w_{ij} is the weight of the input neuron i relatively to the intermediate neuron j, W_{jk} is the weight of the intermediate neuron j relatively to the output neuron k, θ_j is the bias of the intermediate neuron j, Γ_k is the bias of the output neuron k, $f_{\mathrm{I}}(\cdot)$ is the activation function of the intermediate neurons and $f_{\mathrm{Y}}(\cdot)$ is the activation function of the output neurons.

The activation function $f_{\mathrm{I}}(\cdot)$ was defined through the hyperbolic tangent since it allows a faster convergence of the training algorithm [4]. As for the activation function to the output layer, $f_{\mathrm{Y}}(\cdot)$, it is generally a linear function [7]. Mathematically,

$$\text{Hyperbolic tangent}: \quad f_{\mathrm{I}}(q) = \frac{\mathrm{e}^{q} - \mathrm{e}^{-q}}{\mathrm{e}^{q} + \mathrm{e}^{-q}}\,, \qquad -1 < f_{\mathrm{I}} < 1\,, \tag{5}$$

$$\text{Linear}: \quad f_{\mathrm{Y}}(q) = q\,, \qquad -\infty < f_{\mathrm{Y}} < \infty\,, \tag{6}$$

considering q a generic variable.

The continuous and differentiable function with predictive objectives which is generated by the neural network is defined in the vectorial form by

$$\boldsymbol{Y}(\boldsymbol{P}, \boldsymbol{X}) = \boldsymbol{W} \cdot \tanh(\boldsymbol{w}^{\mathsf{T}} \cdot \boldsymbol{X} + \boldsymbol{\theta}) + \boldsymbol{\Gamma}\,, \tag{7}$$

where $\boldsymbol{P}$ represents the set of matrix parameters $\boldsymbol{w} \in \mathbb{R}^{n_{\mathrm{X}} \times n_{\mathrm{I}}}$, $\boldsymbol{W} \in \mathbb{R}^{n_{\mathrm{Y}} \times n_{\mathrm{I}}}$, $\boldsymbol{\theta} \in \mathbb{R}^{n_{\mathrm{I}} \times 1}$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{n_{\mathrm{Y}} \times 1}$; $\boldsymbol{X} \in \mathbb{R}^{n_{\mathrm{X}} \times 1}$ represents the vector of input neurons and $\boldsymbol{Y} \in \mathbb{R}^{n_{\mathrm{Y}} \times 1}$ the vector of output neurons.

Considering a dataset with m points $\{(\boldsymbol{X}_1,\ \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_i,\ \boldsymbol{Y}_i), \ldots, (\boldsymbol{X}_m,\ \boldsymbol{Y}_m)\}$, during the neural network training the parameters are determined so that, for the input $\boldsymbol{X}_i$, the estimate of the output variables $\hat{\boldsymbol{Y}}_i$ should equal the values $\boldsymbol{Y}_i$ [15]. Such optimization problem corresponds to the minimization of the average square error (MSE), that is,

$$\min_{\boldsymbol{P}} \ F(\boldsymbol{P}) = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{e}_i^{\top}\, \boldsymbol{e}_i\,,$$

where $\boldsymbol{e}_i = \boldsymbol{Y}_i - \hat{\boldsymbol{Y}}_i(\boldsymbol{P}, \boldsymbol{X})$. The algorithm of Levenberg-Marquardt was used to solve this problem since it is an efficient algorithm even in cases of strong ill-conditioned problems [13]. The algorithm is based on Gauss-Newton and on Gradient methods and determines $\boldsymbol{P}$ at each iteration using

$$\Delta \boldsymbol{P} = -(\boldsymbol{H} + \mu \boldsymbol{I})^{-1}\, \nabla F\,, \tag{8}$$

where $\boldsymbol{H}$ is the hessian matrix, $\boldsymbol{I}$ is the identity matrix, ∇F is the gradient vector of $F(\boldsymbol{P})$ and μ the learning rate that is updated in order to minimize MSE [10].

4 Dynamic Mathematical Model of a Decanter

Consider now an industrial continuous decanter unit with parallelepipedic format and positioned horizontally, as depicted in Fig. 4.

The decanter input stream is constituted by the mixture that leaves the reactor flowing at a molar rate N_{in} and is characterized by composition $\boldsymbol{x}_{\text{in}}$ and temperature T.

In the decanter, all the components of the feed get split into two phases but in different proportions from component to component. The degree of separation by the two phases for a generic component i is quantified through the split fraction ξ_i which represents the fraction of component i that goes into the light phase. The set of the split fractions to the light phase for all the components is therefore the vector $\boldsymbol{\xi} = \left[\xi_{\text{E}}\ \xi_{\text{M}}\ \xi_{\text{G}}\right]$ and to the heavy phase is its complementary $\mathbf{1} - \boldsymbol{\xi}$.

The decanter is equipped with an internal baffle. As the two phases separate, the heavy phase leaves the unit through its bottom while the light phase leaves the decanter by overflowing the baffle positioned close to its end. The dynamics of the section after the baffle can be neglected since its volume is insignificant compared to the total volume of the decanter. The output molar flow rate of the heavy phase, N_{hv}, is manipulated by a level controller.

A mathematical model describing this system can be developed based on first-principles. With such a purpose, partial and global mass balances were considered. The resulting mechanistic model describes the evolution of the molar fractions of all the components in each of the phases as well as the heights of these phases. For a generic component i ($i = \text{E, M}$),

$$n_{\text{hv}} \frac{\mathrm{d}x_{i,\text{hv}}}{\mathrm{d}t} = \sum_{k}^{n_{\text{c}}} \left((1-\xi_k)\, x_{k,\text{in}}\right) N_{\text{in}} \left(\frac{1-\xi_i}{\sum_{k}^{n_{\text{c}}} \left((1-\xi_k)\, x_{k,\text{in}}\right)} x_{i,\text{in}} - x_{i,\text{hv}} \right) \tag{9}$$

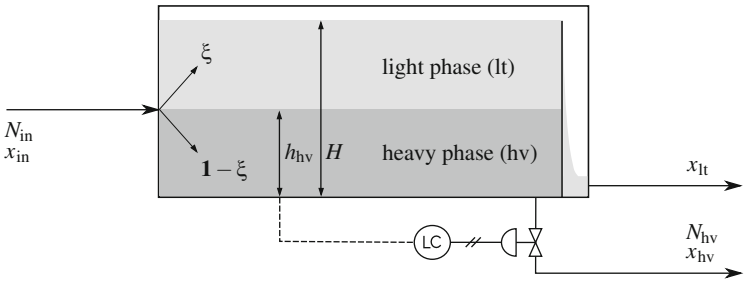


Fig. 4 Schematic representation of the decanter

and

$$n_{\mathrm{lt}} \frac{\mathrm{d}x_{i,\mathrm{lt}}}{\mathrm{d}t} = \sum_{k}^{n_{\mathrm{c}}} (\xi_k \, x_{k,\mathrm{in}}) \; N_{\mathrm{in}} \left(\frac{\xi_i}{\sum_{k}^{n_{\mathrm{c}}} (\xi_k \, x_{k,\mathrm{in}})} x_{i,\mathrm{in}} - x_{i,\mathrm{lt}} \right) , \tag{10}$$

where n_{hv} and n_{lt} represent the molar amount of molecules in the heavy and light phases, respectively. The composition of the remaining component (G) in phase j (with $j = \mathrm{hv}, \mathrm{lt}$) is

$$x_{\mathrm{G},j} = 1 - \sum_{i}^{n_{\mathrm{c}}-1} x_{i,j} \; . \tag{11}$$

From a global molar balance to the heavy phase,

$$\frac{\mathrm{d}n_{\mathrm{hv}}}{\mathrm{d}t} = \sum_{i}^{n_{\mathrm{c}}} ((1 - \xi_i) \, x_{i,\mathrm{in}}) \, N_{\mathrm{in}} \; - \; N_{\mathrm{hv}} \; . \tag{12}$$

The molar amount of molecules in the light phase is given by

$$n_{\mathrm{lt}} = \frac{h_{\mathrm{lt}} \, A}{\sum_{i}^{n_{\mathrm{c}}} (V_i \, x_{i,\mathrm{lt}})} \tag{13}$$

and the heights of both phases by

$$h_{\mathrm{hv}} = \frac{n_{\mathrm{hv}}}{A} \sum_{i}^{n_{\mathrm{c}}} (x_{i,\mathrm{hv}} \, V_i) \; , \tag{14}$$

with

$$h_{\mathrm{lt}} = H - h_{\mathrm{hv}} \; , \tag{15}$$

where A is the area of the base of the decanter and V_i stands for the molar volume of component i.

The split fractions $\boldsymbol{\xi}$ are calculated using the previously developed neural network. Equations 13 and 14 assume that both phases are ideal. The physical properties that constitute model parameters are specified in Table 1.

Table 1 Molar volume of ester, methanol and glycerol

Component	V (10^{-5} m^3 mol^{-1})
E	34.51
M	4.23
G	6.87

5 Results and Discussion

Based on the composition and temperature of a mixture of ester, methanol and glycerol entering the decanter, the neural network must indicate how the three components separate by the light and heavy phases, that is, must predict the split fractions to the light phase for all the components. Thus, from the viewpoint of the neural network, the input variables are the temperature (T) and the composition of the mixture to be separated. The mixture composition is expressed in terms of molar fractions of methanol and of glycerol,[1] $x_{M,in}$ and $x_{G,in}$. The output variables are the split fractions for the three components ξ_E, ξ_M and ξ_G, indicating, for each component, the molar or mass fraction of the initial amount that goes to the light phase. Once known, the split fractions can be used to solve the mathematical model describing the decanter in a CPU time efficient way.

5.1 Generation and Treatment of Data

The liquid-liquid equilibrium data used to train and validate the neural network were generated by the *flash* calculation described in Sect. 2.

The characterization of the initial mixture that enters the separation unit is specially important, since the network must be trained with a set of relevant data with regard to the range usually observed in such systems. The authors of [1] experimentally performed the transesterification reaction of sunflower oil at 60 °C using a molar ratio between methanol and oil of 6:1, 0.50 % (*m*/*m*) of NaOH as catalyst and an agitation rate of 400 rpm. In that work, the molar ratio of the component concentration over time is shown. However, the information about methanol, one of the components in largest quantity in the mixture, is omitted. For this reason, it was necessary to simulate the transesterification reaction (reactor) in order to obtain the dynamic profiles of the composition of different chemical species needed to completely quantify the mixture at the end of the reaction, in particular with regard to methanol. The equilibrium and the speed of all the transesterification reactions are conditioned by reaction medium stirring. Work [5] proposes a methodology to systematically include this variable in the model of the

[1] Note that the molar fraction of the ester is linearly dependent of the molar fractions of the two other components.

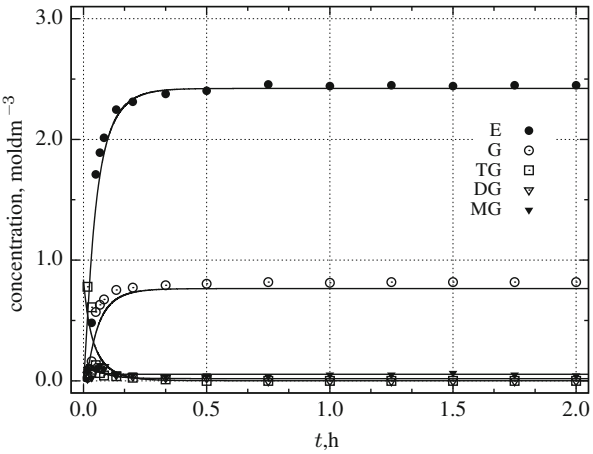


Fig. 5 Simulation of the transesterification reaction of sunflower oil

reactor. However, for the purposes of the present work, a more simplistic model is enough to generate the data sets. The model and parameters described in the work [1] were used in the simulation system and the corresponding results are shown in Fig. 5. The visual comparison between simulated and experimental points as well as the coefficient of determination ($R^2 = 0.99998$) allow to conclude on the good reproduction of the system studied in [1].

The exhausted reaction mixture that leaves the reactor is then directed to the decanter without undergoing any change in its composition. Therefore, the simulation values of the reactor for the final time ($t = 2$ h) correspond to the concentration values at the entrance of the decanter (mixture before separation). The molar concentration of the mixture to be separated is $\boldsymbol{C} = \left[C_{\mathrm{TG}}\ C_{\mathrm{DG}}\ C_{\mathrm{MG}}\ C_{\mathrm{M}}\ C_{\mathrm{G}}\ C_{\mathrm{E}}\right] = \left[0.0018\ 0.0188\ 0.0550\ 2.6181\ 0.7644\ 2.4219\right]$ mol dm^{-3}.

The fractions of tri-, di- and monoglycerides were considered to be in the fraction of the ester component since their amounts are reduced and that molecules present similar properties. The corresponding molar fraction is given by $\boldsymbol{x} = \left[x_{\mathrm{E}}\ x_{\mathrm{M}}\ x_{\mathrm{G}}\right] \approx \left[x_{\mathrm{E}} + x_{\mathrm{TG}} + x_{\mathrm{DG}} + x_{\mathrm{MG}}\ x_{\mathrm{M}}\ x_{\mathrm{G}}\right] = \left[0.42\ 0.45\ 0.13\right]$.

To generate the experimental data pertaining to the liquid-liquid equilibrium, various temperatures of the input flow were used. For each temperature, a mesh was constructed by varying the molar fractions of the mixture. The intervals considered were: $25 < T < 60$°C, $0.32 < x_{\mathrm{E,in}} < 0.52$ and $0.35 < x_{\mathrm{M,in}} < 0.55$. The molar fraction of glycerol at the entrance of the decanter, $x_{\mathrm{G,in}}$, was computed by the relation $\sum_{i=1}^{n_c} x_{i,\mathrm{in}} = 1$. The range for the temperature was selected taking into account the typical reaction temperatures defined by [1]. This range was then covered with increments of 1°C. The range of compositions was defined taking into account a range of ± 0.10 in molar fractions $x_{\mathrm{E,in}}$ and $x_{\mathrm{M,in}}$ previously calculated. The defined

Table 2 Average (μ) and standard deviation (σ) for the normalization of the input data

Input data	μ	σ
$x_{M,in}$, – (n/n)	0.44291	0.05769
$x_{G,in}$, – (n/n)	0.14301	0.07483
T, °C	42.34	10.39

range for compositions was scanned through increments of 0.01. In total, 36 meshes of 405 points were generated.

The normalization of data plays a key role in training the neural network. The use of data from different orders of magnitude can favor the attribution of different adjustment importances during the training phase of the neural network [7], whereupon the data was normalized using the values given in Table 2.

After the pre-treatment, data were randomly divided into three sets: the training set, the validation set and the test set. The training and validation sets were used to adjust the neural network. The test set was used to simulate the network allowing further comparison between the data obtained by the method *flash* and the predicted by the neural network.

5.2 Characterization of the Neural Network

The neural network was structured into three distinct layers. The input layer has three neurons corresponding to the three input variables $\boldsymbol{X} = \left[x_{M,in}\ x_{G,in}\ T\right]^{\top}$. An intermediate layer having five neurons and an output layer with three neurons corresponding to the variables $\boldsymbol{Y} = \left[\xi_E\ \xi_M\ \xi_G\right]^{\top}$ are considered. Figure 3 graphically depicts the network structure.

The training algorithm described in Sect. 3 is available in the software package `octave-nnet 0.1.13-2` for GNU Octave [12] and was used in training the neural network. In this algorithm, the weights initialization is made using random elements uniformly distributed in the interval $[-1, 1]$. The initial learning rate μ_0 is set to 10^{-3} [10]. Other parameters related to the training of the neural network were specified as follows: the maximum number of iterations was 2×10^3, the tolerance was 5×10^{-7} and the maximum time for training was 10^3 s.

5.3 Neural Network Training

The neural network with the described structure was trained. Figure 6 shows (points) the evolution of the mean squared error over the training iterations. The Levenberg-Marquardt algorithm took 99 iterations to achieve the specified tolerance of 5×10^{-7} (dashed line). The training process took 20 s. The network validation was done automatically by the software package used. Figure 6 compares the MSE

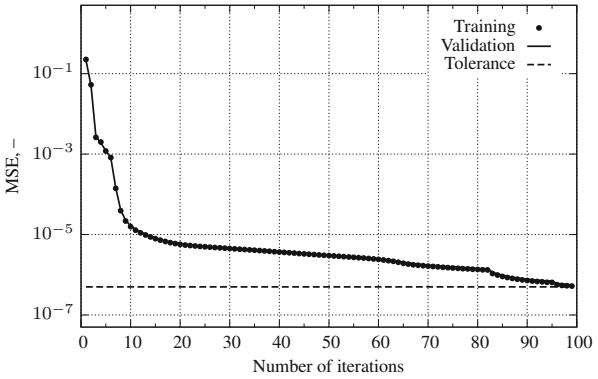


Fig. 6 Evolution of the average square error while applying the Levenberg-Marquardt algorithm

of the validation along iterations (solid line) with the MSE of the training dataset, being possible to see that they are coincident.

The neural network training allows the determination of the weighting matrices $\boldsymbol{w}$ and $\boldsymbol{W}$ defined by

$$\boldsymbol{w} = \begin{bmatrix} -0.280399 & -1.089354 & -0.085569 & 0.139296 & 0.074241 \\ -0.133304 & -0.569687 & -0.994122 & 0.322463 & 1.155133 \\ -0.477709 & 0.281168 & 0.030668 & -0.061436 & -0.017357 \end{bmatrix}$$

and

$$\boldsymbol{W} = \begin{bmatrix} 1.6266 \times 10^{-4} & 1.7624 \times 10^{-4} & 8.4364 \times 10^{-3} & 1.2702 \times 10^{-4} & 4.2978 \times 10^{-3} \\ 5.7984 \times 10^{-3} & 7.9335 \times 10^{-3} & 2.2665 \times 10^{-0} & -2.9864 \times 10^{-1} & 7.3118 \times 10^{-1} \\ -1.3643 \times 10^{-3} & 7.3683 \times 10^{-4} & 3.9128 \times 10^{-1} & 2.6466 \times 10^{-3} & 1.5635 \times 10^{-1} \end{bmatrix}$$

and the bias vectors $\boldsymbol{\theta}$ and $\boldsymbol{\Gamma}$ corresponding to

$$\boldsymbol{\theta} = \begin{bmatrix} 0.62280 \\ 1.15884 \\ -2.46359 \\ 0.27300 \\ 2.24627 \end{bmatrix} \text{ and } \boldsymbol{\Gamma} = \begin{bmatrix} 1.00373 \\ 1.89116 \\ 0.23502 \end{bmatrix}.$$

Since the objective of the work was the development of a neural network that could replace in a faster but equally effective way the traditional *flash* calculation, measures of average computation times required by each method were registered. The average time required to generate a point by the *flash* calculation was about 0.018 129 s, while using the neural network was approximately 140 times smaller. The time used by the neural network was substantially less than the time required

by the *flash* calculation (about 141 times smaller), showing the appropriateness of the choice made for the application in question.

5.4 Predicting Capability of the Network

Figure 7 shows the prediction of the split fractions using the neural network. It also includes the first 250 points of the test.

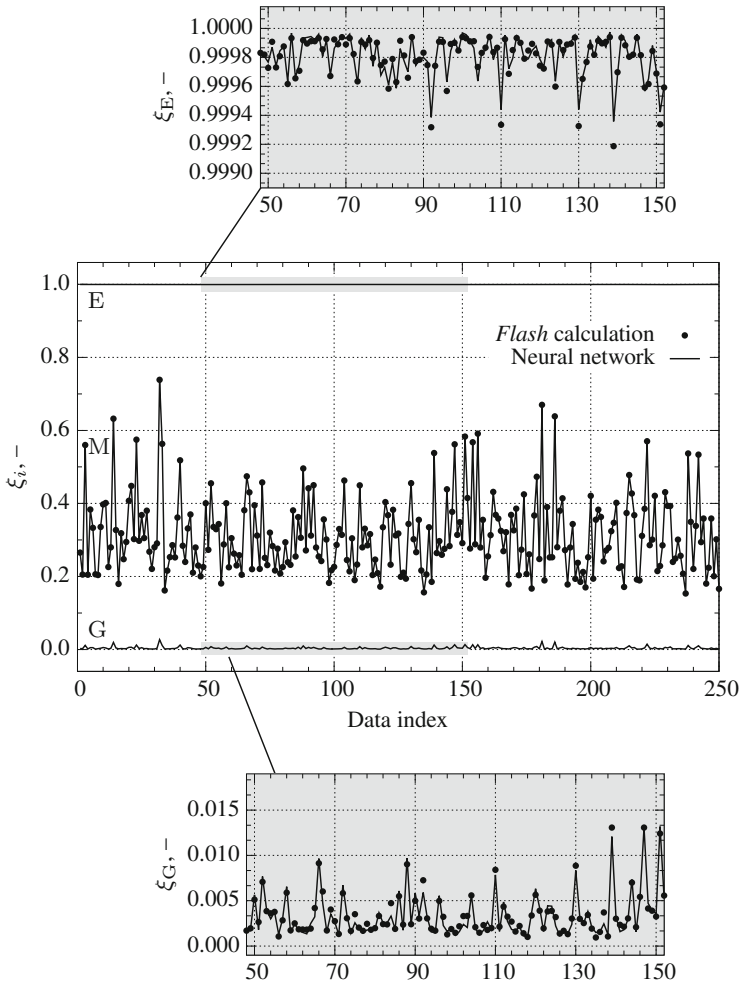


Fig. 7 Prediction of the split fractions through the neural network model

Methanol is the component with the biggest variation of its split fraction within the range of temperature and composition covered by the mesh. As for the ester, it is the component for which the split fraction is less dependent on the initial conditions of the mixture to be treated (i.e., its composition and temperature). In fact, as it can be seen in Fig. 7, more than 99.9 % of the ester always goes to the light phase, regardless of the initial conditions of the mixture to be separated. Finally, the variation of the split fraction of glycerol as a function of composition and temperature of the decanter feeding mixture is also slight. Glycerol migrates almost entirely to the heavy phase. To allow for a better understanding of the data, in Fig. 7, two areas of the main graph were zoomed out, one relative to the data for the glycerol component and other to data concerning the ester component.

The determination coefficients corresponding to the estimates of the components methanol, ester and glycerol are, respectively, $R^2(\xi_M) = 0.9999$, $R^2(\xi_E) = 0.9137$ and $R^2(\xi_G) = 0.9676$. The prediction is especially good in the case of methanol, since this component is more sensitive to the initial conditions of the mixture. However, although in the case of glycerol and of ester the determination coefficients are somewhat lower, it is noteworthy that the absolute errors between the predictions and the experimental values are extremely reduced (see Fig. 8).

The effect of temperature on the liquid-liquid equilibrium is very important. To show the predictive capability of this effect by the neural network, two equilibria corresponding to two mixtures A and B with different compositions under different temperatures were studied.

Mixture A is characterized by a molar fraction $\boldsymbol{x}_{in} = [0.42\ 0.45\ 0.13]$, values consistent with the experimental values of [1]. The second study deals with a mixture, designated B, which represents a mixture resulting for a greater extent of chemical reaction, i.e., represents a reaction mixture originated in a situation of greater yield than the one verified when mixture A was originated. Based on this assumption, mixture B was defined as having the composition $\boldsymbol{x}_{in} = [0.47\ 0.43\ 0.10]$.

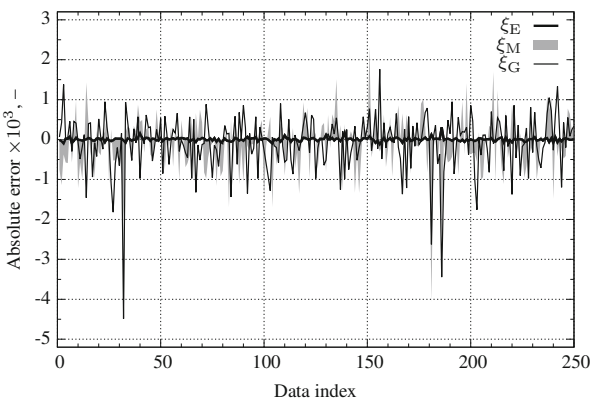


Fig. 8 Absolute error between the predictions of the neural network and the of the *flash* calculation

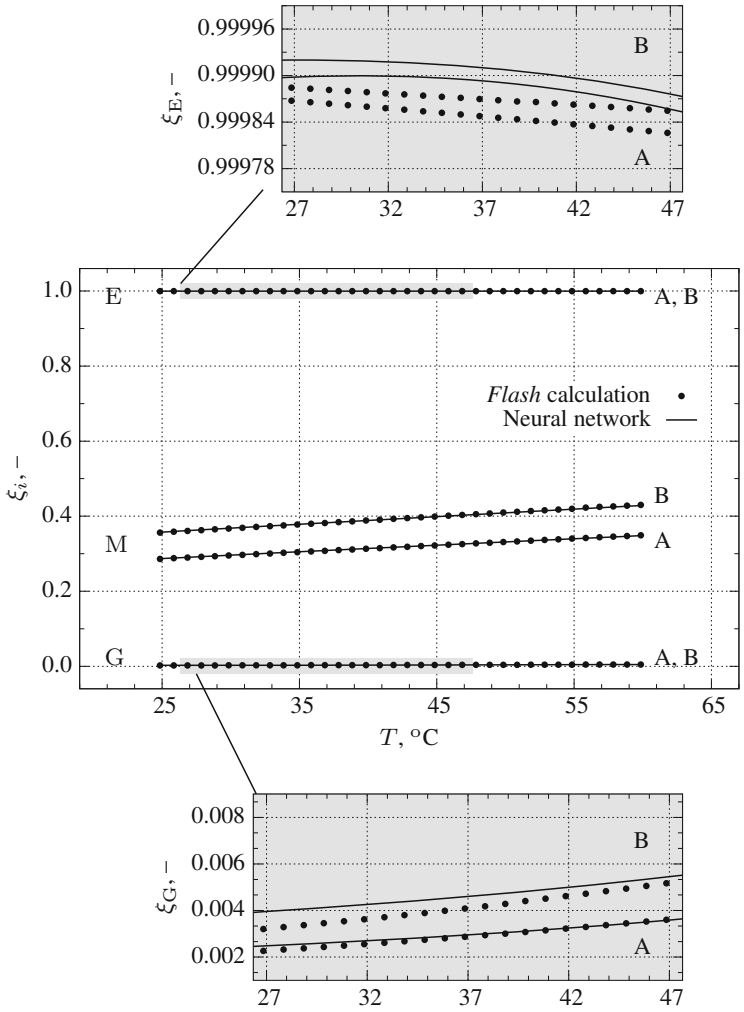


Fig. 9 Prediction of the split fractions as function of temperature for two different mixtures A and B (A: $[0.42\ 0.45\ 0.13]$, B: $\boldsymbol{x}_{\text{in}} = [0.47\ 0.43\ 0.10]$)

Figure 9 represents the split fractions predicted by the *flash* calculation and by the neural network as functions of the temperature of the mixtures, being evident a good prediction of the neural network.

As discussed above, the split fraction for methanol varies significantly with temperature, in opposition to the fractions of ester and glycerol that remain approximately constant. A considerable zoom out of the graphical representation of these two fractions (see Fig. 9) reveals what, at a first glance, could be considered as a discrepancy, particularly in the case of the ester. However, keep in mind that

the "discrepancy" is less than 0.006 % (6 thousandths percent), and is therefore negligible.

Increasing the temperature, the methanol and the glycerol split fractions increase in both mixtures, although with less intensity in the case of glycerol. In mixture B (mixture richer in ester), methanol is more soluble in the light phase and, therefore, the split fraction is greater than the one obtained to mixture A. Similar effect is observed for glycerol. Conversely, by increasing the temperature of the mixture, the ester becomes more soluble in the heavy phase and therefore the split fraction to the light phase for ester decreases.

5.5 *Application in a Dynamic Decanter*

The artificial neural network presented above quantifies the interaction between the two liquid phases by calculating the split fractions for all the components. Because of its computational advantages over the iterative *flash* calculations, the network is applied herein as part of the dynamic model of a decanter developed in Sect. 4.

Suppose that the continuous decanter with dimensions 1 m×1 m×3 m is, at initial time, completely filled: half with glycerol and half with ester. This combination forms two immiscible liquid phases with glycerol at the lower layer due to its higher density. Therefore, the initial height of the heavy phase is $h_{\mathrm{hv}} = 0.5\,\mathrm{m}$ and the initial height of the light phase is $h_{\mathrm{lt}} = 0.5\,\mathrm{m}$. In such conditions, the initial phase compositions are $\boldsymbol{x}_{\mathrm{lt}} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$ and $\boldsymbol{x}_{\mathrm{hv}} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$. At the same initial instant, the reaction mixture is fed to the decanter with a flow rate of $N_{\mathrm{in}} = 9.67\,\mathrm{mol\,s^{-1}}$, composition $\boldsymbol{x}_{\mathrm{in}} = \begin{bmatrix}0.42 & 0.45 & 0.13\end{bmatrix}$ (corresponding to the aforementioned mixture A), and temperature $T = 60\,^{\circ}\mathrm{C}$.

The heavy phase level h_{hv} is controlled through a PI(D) controller using the molar flow rate N_{hv} as manipulated variable (initialized at $0\,\mathrm{mol\,s^{-1}}$). The controller was tuned by the trial-and-error method with $K_{\mathrm{C}} = -500\,\mathrm{mol\,s^{-1}\,m^{-1}}$, $\tau_{\mathrm{I}} = 2000\,\mathrm{s}$, and $\tau_{\mathrm{D}} = 0\,\mathrm{s}$.

5.5.1 Operation Start-Up

The decanter start-up operation is simulated along a time horizon of 20 h and using a time interval of 10 s. Figure 10 exhibits the dynamic response of the unit. As soon as the feed is introduced, the composition of the light and of the heavy phases change due to the entrance of new components.

The split fractions computed by the neural network allow to define the affinity that each component will have to each of the heavy and light phases. For the feed conditions listed above, the split fractions are $\boldsymbol{\xi} = \begin{bmatrix}0.9998 & 0.3481 & 0.0046\end{bmatrix}$. Remark the high split fraction to the light phase for ester and the low split fraction for glycerol.

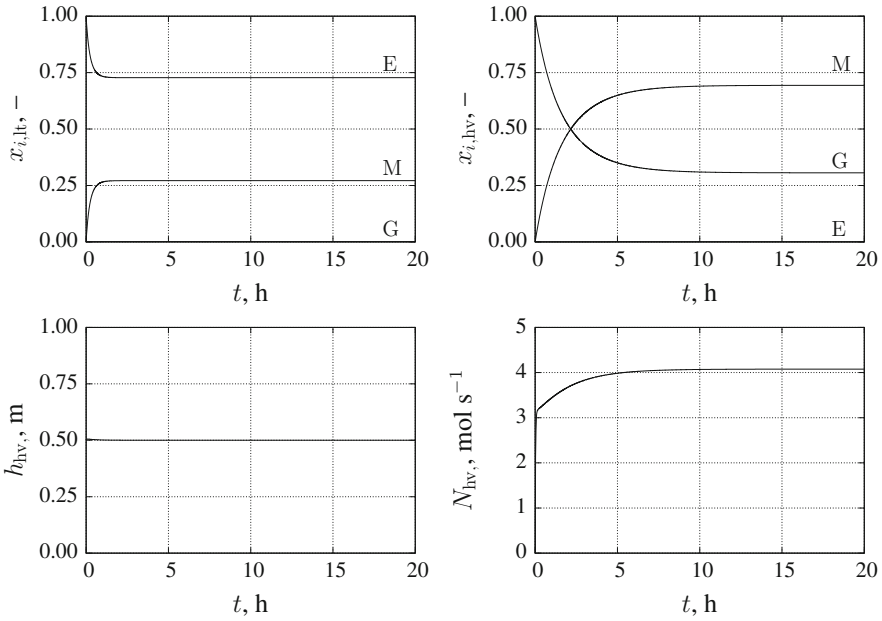


Fig. 10 Profiles of the state variables and molar flow rate of the heavy phase under the start-up of the decanter operation

Methanol is attracted by both phases originating changes in the composition of both phases in what concerns this component. Glycerol does not have much affinity to the light phase and, as result, its molar fraction remains near zero in this phase. Conversely, the ester goes almost exclusively to the light phase, reason why the composition of the heavy phase in ester keeps approximately zero. This behavior is determined by the split fractions calculated by the neural network previously trained.

After approximately 10 h, the decanter reaches a steady-state with a composition of the light phase $\boldsymbol{x}_{\mathrm{lt}} = \begin{bmatrix}0.728 & 0.271 & 0.001\end{bmatrix}$ and a composition of the heavy phase $\boldsymbol{x}_{\mathrm{hv}} = \begin{bmatrix}0.000 & 0.694 & 0.306\end{bmatrix}$. From the graphs of Fig. 10 it is also evident that the light phase presents a much faster dynamics than the heavy phase. In spite of the volumes of both phases to be the same throughout the experiment (0.5 m^3), the light phase is crossed by a significantly higher volumetric flow (bigger than 7 times) when compared to the volumetric flow crossing the heavy phase. In consequence, the residence time in the light phase is much smaller resulting in a faster dynamic response.

As it is clear from Fig. 10, the level is kept by the controller at the set-point of 0.5 m during the whole test. To maintain this value, the controller increases the output molar flow rate N_{hv} from zero until it finally stabilizes at 4.08 mol s^{-1}.

5.5.2 Effect of Disturbances

The importance of the liquid-liquid equilibrium description is further underlined with the analysis of the decanter under the effect of disturbances. The system, reinitialized at the steady-state encountered during the study of the system start-up, is subjected to various disturbances at instants $t = 6\,\text{h}$ and $t = 18\,\text{h}$. Figure 11 reveals the evolution of the key variables describing the system behavior in such situations. The ester composition in the heavy phase and the glycerol composition in the light phase were omitted from the graphs because they remained very low (approximately zero) throughout the whole test.

At instant $t = 6\,\text{h}$, the mixture that constitutes the feed is replaced by a mixture richer in ester (that is, the feed is changed from mixture A to mixture B). Therefore, the feed composition changes to $\boldsymbol{x}_{\text{in}} = \left[0.47\ 0.43\ 0.10\right]$. In view of this new condition, the neural network foresees a new liquid-liquid equilibrium and, in accordance, updates the split fractions to $\boldsymbol{\xi} = \left[0.9998\ 0.4283\ 0.0069\right]$. It is worth mentioning that the ester and glycerol split fractions for the light phase do not suffer significant changes. However, the methanol split fraction increases substantially.

Mixture B is poorer in methanol than mixture A. This reduces the amount of methanol going to both phases inside the decanter. However, the new feed originates, in parallel, a bigger split fraction to the light phase for methanol. A bigger methanol

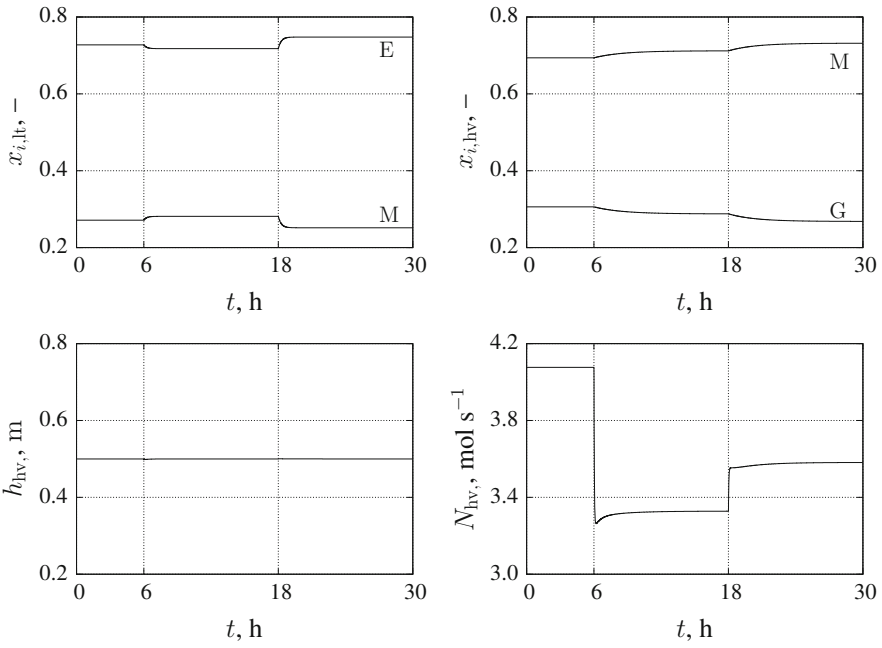


Fig. 11 Profiles of the state variables and molar flow rate of the heavy phase under disturbances to the operation

split fraction induces a bigger amount of methanol going to the light phase. This second effect overlaps the first and, in consequence, the amount of methanol going to the light phase increases as a result of the disturbance introduced at $t = 6\,\text{h}$. The total molar amount moving into the light phase also increases as a consequence of this disturbance (because of methanol but, especially, because of ester). Although this fact tends to reduce the molar fraction, the increase in the amount of methanol is enough to impose an increase in methanol molar fraction, as shown in Fig. 11. Therefore, the amount of methanol going to the heavy phase decreases as a result of the introduced disturbance. However, since the total molar amount going to the heavy phase decreases (because of smaller methanol and glycerol contributions), the molar fraction of methanol increases as Fig. 11 reveals.

The amount of ester going to the light phase increases, but its molar fraction decreases due to the more significant effect of the overall amount increase in the light phase (namely methanol and ester). In what concerns the molar fraction of glycerol in the heavy phase, it diminishes (see Fig. 11). On one hand the amount of this component passing to the heavy phase is less and, on the other, the total molar amount of the heavy phase is higher.

To keep the level at the set-point, the flow rate N_{hv} is changed. Once the rates of methanol and glycerol sent to the heavy phase are smaller, the controller has to lower the flow N_{hv} in order to be able to maintain the level at its set-point.

After having reached a steady-state, at $t = 18\,\text{h}$ the feed temperature is reduced from 60°C to 30°C. This disturbance changes again the component distribution (the split fraction becomes $\boldsymbol{\xi} = \begin{bmatrix} 0.9999 & 0.3677 & 0.0041 \end{bmatrix}$). For this new operating conditions, the fraction of the inlet methanol that goes to the heavy phase is higher, inducing an increase of the methanol molar fraction and a larger glycerol dilution, that is, a decrease in glycerol molar fraction in the heavy phase.

At the same time, the methanol molar fraction to the light phase decreases. Consequently, a smaller rate of methanol is directed to this phase whilst the molar rates of the other two components remain practically unchanged. Therefore, the molar fraction of methanol and ester in the light phase augments and diminishes, respectively.

The level controller increases again the flow rate since the rate of methanol sent to the heavy phase has also increased as a result of this second disturbance.

6 Conclusions

Process dynamic simulation and model based control of a phase separator in biodiesel industry require a mechanistic dynamic model of the unit and, therefore, a way of quantifying the liquid-liquid equilibrium. The *flash* calculation typically used to describe the liquid-liquid equilibrium is inadequate in this situation because it is an iterative method. Thus, it is suggested here to approximate the *flash* calculation by an artificial neural network of the type *feedforward back-propagation*

that predicts the separation of the initial ternary mixture composed of ester, methanol and glycerol as a function of the compositions and the temperature.

With this approach, an iterative calculation subject to a stop condition based on comparison of adjacent predictions was avoided. Moreover, it enables the use of automatic differentiation tools to facilitate the resolution of nonlinear optimization problems. These advantages were achieved without jeopardizing the quality of the global model since the predictions of the split fractions obtained by the neural network model reproduce well the equilibrium data obtained by the *flash* calculation. Additionally, the computation time was significantly reduced with the use of the neural network by avoiding the typical iterative process of the *flash* calculation.

In order to investigate the impact of the phase equilibrium described by a neural network on the model of the decanter (needed for example for optimization or for model control of the unit) a set of dynamic simulations were conducted. A first-principle model was used to predict the dynamic behavior of the system in closed loop. The on-line results from the developed network were incorporated in the model. It was possible to describe the decanter behavior in a computationally effective way, compatible with objectives of dynamic optimization and model based control of the real unit.

Acknowledgements This work had financial support from QREN through Operational Programme Mais Centro and from the European Union via FEDER under APCFAME project with reference 3509/2009, a consortium between Ciengis, SA and the University of Coimbra. The authors also express their thanks to Vitor Marques for access to his preliminary studies on equilibrium data.

References

1. Bambase, M., Nakamura, N., Tanaka, J., Matsumura, M.: Kinetics of hydroxide-catalyzed methanolysis of crude sunflower oil for the production of fuel-grade methyl esters. J. Chem. Technol. Biotechnol. **82**(3), 273–280 (2007). doi:10.1002/jctb.1666
2. Bell, B.M.: CppAD: a package for C++ algorithmic differentiation. Computational Infrastructure for Operations Research **COIN-OR** (http://www.coin-or.org/CppAD) (2012)
3. Biegler, L.: Real-Time PDE-Constrained Optimization. Computational Science and Engineering. Society for Industrial and Applied Mathematics, Philadelphia (2007)
4. Bishop, C.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)
5. Brásio, A.S., Romanenko, A., Santos, L.O., Fernandes, N.C.: Modeling the effect of mixing in biodiesel production. Bioresour. Technol. **102**(11), 6508–6514 (2011). doi:10.1016/j.biortech.2011.03.090
6. Brásio, A.S., Romanenko, A., Leal, J., Santos, L.O., Fernandes, N.C.: Nonlinear model predictive control of biodiesel production via transesterification of used vegetable oils. J. Process Control **23**(10), 1471–1479 (2013). doi:10.1016/j.jprocont.2013.09.023
7. Chaturvedi, D.: Soft Computing: Techniques and Its Applications in Electrical Engineering. Studies in Computational Intelligence. Springer, Berlin/Heidelberg (2008)
8. Du, X., Liu, L., Xi, X., Yang, L., Yang, Y., Liu, Z., Zhang, X., Yu, C., Du, J.: Back pressure prediction of the direct air cooled power generating unit using the artificial neural network model. Appl. Therm. Eng. **31**(14–15), 3009–3014 (2011). doi:10.1016/j.applthermaleng.2011.05.034

9. Fredenslund, A., Jones, R.L., Prausnitz, J.M.: Group-contribution estimation of activity coefficients in nonideal liquid mixtures. AIChE J. **21**(6), 1086–1099 (1975). doi:10.1002/aic.690210607
10. Hagan, M., Demuth, H., Beale, M.: Neural Network Design. Electrical Engineering Series. Brooks/Cole, Boston (1996)
11. Lobo, L.Q., Ferreira, A.G.M.: Termodinâmica e Propriedades Termofísicas – Volume I: Termodinâmica das Fases. Imprensa da Universidade de Coimbra, Coimbra (2006)
12. Schmid, M.D.: A neural network package for Octave – User's Guide (2009). http://www.plexso.com/61_octave/neuralNetworkPackageForOctaveDevelop.pdf. Consulted in March 2013
13. Sjöberg, J.: Neural Networks – Train and analyze neural networks to fit your data. Technical report, Wolfram Research (2005). http://media.wolfram.com/documents/NeuralNetworksDocumentation.pdf. Consulted in March 2013
14. Walther, A., Griewank, A.: Getting started with ADOL-C. In: Naumann, U., Schenk, O. (eds.) Combinatorial Scientific Computing. Chapman-Hall CRC Computational Science, chap. 7, pp. 181–202. CRC Press, Boca Raton (2012)
15. Yegnanarayana, B.: Artificial Neural Networks. Prentice-Hall Of India Pvt. Limited, New Delhi (2004)

Determination of (0, 2)-Regular Sets in Graphs and Applications

Domingos M. Cardoso, Carlos J. Luz, and Maria F. Pacheco

Abstract In this paper, relevant results about the determination of (κ, τ)-regular sets, using the main eigenvalues of a graph, are reviewed and some results about the determination of $(0, 2)$-regular sets are introduced. An algorithm for that purpose is also described. As an illustration, this algorithm is applied to the determination of maximum matchings in arbitrary graphs.

1 Introduction

All graphs considered throughout this paper are simple (with no loops nor multiple edges), undirected and have order n. $V(G) = \{1, 2, \ldots, n\}$ and $E(G)$ denote, respectively, the vertex and the edge sets of G and ij represents the edge linking nodes i and j of $V(G)$. If $i \in V(G)$, then the vertex set denoted by $N_G(i) = \{j \in V(G) : ij \in E(G)\}$ is called neighbourhood of i. Additionally, $N_G[i]$ denotes the closed neighbourhood of vertex i (that is, $N_G[i] = N_G(i) \cup \{i\}$). Given a graph G and a set of vertices $U \subset V(G)$, the subgraph of G induced by $U, G[U]$, is such that $V(G[U]) = U$ and $E(G[U]) = \{ij : i, j \in U \wedge ij \in E(G)\}$. A (κ, τ)-*regular set* of a graph is a vertex subset inducing a $\kappa-$regular subgraph such that every vertex not in the subset has τ neighbours in it, [2].

D.M. Cardoso (✉)
CIDMA – Centro de Investigação e Desenvolvimento em Matemática e Aplicações, Departamento de Matemática, Universidade de Aveiro, Aveiro, Portugal
e-mail: dcardoso@ua.pt

C.J. Luz
CIDMA – Centro de Investigação e Desenvolvimento em Matemática e Aplicações, Universidade de Aveiro, Aveiro, Portugal
e-mail: carlos.luz@ua.pt

M.F. Pacheco
CIDMA – Centro de Investigação e Desenvolvimento em Matemática e Aplicações, Universidade de Aveiro, Aveiro, Portugal

Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Bragança, Bragança, Portugal
e-mail: pacheco@ipb.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_7

The adjacency matrix $A_G = [a_{ij}]$ of G is the $n \times n$ symmetric matrix such that $a_{ij} = 1$ if $ij \in E(G)$ and $a_{ij} = 0$ otherwise. The n eigenvalues of A_G are usually called the eigenvalues of G and are ordered $\lambda_{max}(G) = \lambda_1(G) \geq \cdots \geq \lambda_n = \lambda_{min}(G)$. These eigenvalues are all real because A_G is symmetric. It is also known that, provided G has at least one edge, we have that $\lambda_{min}(G) \leq -1$ and, furthermore, $\lambda_{min}(G) = -1$ if and only if every connected component of G is complete, [4]. The multiplicity of λ_i as eigenvalue of G (and, consequently, as eigenvalue of A_G) is denoted by $m(\lambda_i)$. Throughout this paper, $\sigma(G)$ will denote the spectrum of G, that is, the set of G's eigenvalues together with their multiplicities. The eigenspace associated to each eigenvalue λ of G is denoted by $\mathscr{E}_G(\lambda)$.

An eigenvalue of a graph G is *main* if its associated eigenpace is not orthogonal to the all-one vector $\mathbf{j}$. The vector space spanned by such eigenvectors of G is denoted $Main(G)$. The remaining (distinct) eigenvalues of G are referred to as *non-main*. The dimension of $\mathscr{E}_G(\lambda_i)$, the eigenspace associated to each main eigenvalue λ_i of G, is equal to the multiplicity of λ_i. The *index* of G, its largest eigenvalue, is main. The concepts of main and non-main eigenvalue were introduced in [4]. An overview on the subject was published in [5].

Given a graph G, the *line graph* of G, which is denoted by $L(G)$, is constructed by taking the edges of G as vertices of $L(G)$ and joining two vertices in $L(G)$ by an edge whenever the corresponding edges in G have a common vertex. The graph G is called the *root graph* of $L(G)$.

A *stable set* (or independent set) of G is a subset of vertices of $V(G)$ whose elements are pairwise nonadjacent. The stability number (or independence number) of G is defined as the cardinality of a largest stable set and is usually denoted by $\alpha(G)$. A maximum stable set of G is a stable set with $\alpha(G)$ vertices. Given a nonnegative integer k, the problem of determining whether G has a stable set of size k is *NP*-complete and, therefore, the determination of $\alpha(G)$ is, in general, a hard problem.

A *matching* in a graph G is a subset of edges, $M \subseteq E(G)$, no two of which have a common vertex. A matching with maximum cardinality is called a maximum matching. Furthermore, if for each vertex $i \in V(G)$ there is one edge of the matching M incident with i, then M is called a perfect matching. It is obvious that every perfect matching is also a maximum matching. Notice that the determination of a maximum stable set of a line graph, $L(G)$, is equivalent to the determination of a maximum matching of G. There are several polynomial-time algorithms for the determination of a maximum matching of a graph.

The present paper introduces an algorithm for the recognition of (0,2)-regular sets in graphs and the application of this algorithm is illustrated with the determination of maximum matchings through an approach involving (0,2)-regular sets.

2 Main Eigenvalues, Walk Matrix and (κ, τ)-Regular Sets

We begin this section recalling a few concepts and surveying some relevant results.

If G has p distinct main eigenvalues $\mu_1, \ldots, \mu_p$, the main characteristic polynomial of G is

$$m_G(\lambda) = \lambda^p - c_o\lambda^{p-1} - c_1\lambda^{p-2} - \cdots - c_{p-2}\lambda - c_{p-1} = \prod_{i=1}^{p}(\lambda - \mu_i).$$

Theorem 1 ([5]) *If G is a graph with p main distinct eigenvalues $\mu_1, \ldots, \mu_p$, then the main characteristic polynomial of G, $m_G(\lambda)$, has integer coefficients.*

Considering A_G, the adjacency matrix of graph G, the entry $a_{ij}^{(k)}$ of A^k is the number of walks of length k from i to j. Therefore, the $n \times 1$ vector $A^k\mathbf{j}$, gives the number of walks of length k starting in each vertex of G. Given a graph G of order n, the $n \times k$ walk matrix of G is the matrix $W_k = (\mathbf{j}, A\mathbf{j}, A^2\mathbf{j}, \ldots, A^{k-1}\mathbf{j})$. If G has p distinct main eigenvalues, the $n \times p$ walk matrix

$$W = W_p = (\mathbf{j}, A\mathbf{j}, A^2\mathbf{j}, \ldots, A^{p-1}\mathbf{j})$$

is referred to as the *walk matrix* of G. The vector space spanned by the columns of W is called $Main(G)$ and it coincides with the vector space spanned by $v_1, \ldots, v_p$ with $v_i \in \mathscr{E}_G(\mu_i)$ and $v_i^t\mathbf{j} \neq 0, i = 1, \ldots, p$. The orthogonal complement of $Main(G)$ is denoted, as expected, $Main(G)^\perp$. Notice that both $Main(G)$ and $Main(G)^\perp$ are invariant under A_G.

Taking into account that

$$m_G(A_G) = 0 \Leftrightarrow A_G^p\mathbf{j} - c_0A_G^{p-1}\mathbf{j} - c_1A_G^{p-2}\mathbf{j} - \cdots - c_{p-2}A_G\mathbf{j} - c_{p-1}\mathbf{j} = 0, \qquad (1)$$

the following result holds.

Theorem 2 ([3]) *If G has p main distinct eigenvalues, then*

$$W\begin{pmatrix} c_{p-1} \\ \vdots \\ c_1 \\ c_0 \end{pmatrix} = A^p\mathbf{j},$$

where c_j, with $0 \leq j \leq p-1$, are the coefficients of the main characteristic polynomial of G.

It follows from this theorem that the coefficients of the main characteristic polynomial of a graph can be determined solving the linear system

$$Wx = A^p\mathbf{j}.$$

Proposition 1 ([2]) *A vertex subset S of a graph G with n vertices is (κ, τ)-regular if and only if its characteristic vector is a solution of the linear system*

$$(A_G - (\kappa - \tau)I_n)x = \tau \mathbf{j}. \tag{2}$$

It follows from this result that if system (2) has a $(0, 1)$-solution, then such solution is the characteristic vector of a (κ, τ)-regular set. In fact, let us assume that x is a $(0, 1)$-solution of (2). Then, for all $i \in V(G)$,

$$(A_G)_i x = |N_G(i) \cap S| = \begin{cases} \kappa \text{ if } i \in S \\ \tau \text{ if } i \notin S \end{cases}.$$

Next, we will associate to each graph G and each pair of nonnegative numbers, (κ, τ), the following parametric vector, [3]:

$$\mathbf{g}_G(\kappa, \tau) = \sum_{j=0}^{p-1} \alpha_j A_G^j \mathbf{j}, \tag{3}$$

where p is the number of distinct main eigenvalues of G and $\alpha_0, \ldots, \alpha_{p-1}$ is a solution of the linear system:

$$\begin{pmatrix} \kappa - \tau & 0 & \ldots & 0 & -c_{p-1} \\ -1 & \kappa - \tau & \ldots & 0 & -c_{p-2} \\ 0 & -1 & \ldots & 0 & -c_{p-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & -1 & \kappa - \tau - c_0 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{p-2} \\ \alpha_{p-1} \end{pmatrix} = -\tau \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{4}$$

The following theorem is a slight variation of a result proven in [3].

Theorem 3 ([3]) *Let G be a graph with p distinct main eigenvalues $\mu_1, \ldots, \mu_p$. A vertex subset $S \subset V(G)$ is (κ, τ)-regular if and only if its characteristic vector $x(S)$ is such that*

$$x(S) = \boldsymbol{g} + \boldsymbol{q},$$

with

$$\boldsymbol{g} = \sum_{j=0}^{p-1} \alpha_j A^j \mathbf{j},$$

$(\alpha_0, \ldots, \alpha_{p-1})$ is the unique solution of the linear system (4) and if $(\kappa - \tau) \notin \sigma(G)$ then $\boldsymbol{q} = 0$ else $\boldsymbol{q} \in \mathscr{E}_G(\kappa - \tau)$ and $\kappa - \tau$ is non-main.

3 Main Results

An algorithm for the recognition of $(0, 2)$-regular sets in general graphs is introduced in this section. Such algorithm is not polynomial in general and its complexity depends on the multiplicity of -2 as an eigenvalue of the adjacency matrix of A_G. Particular cases for which the application of the algorithm is polynomial are presented.

Theorem 4 *If a graph G has a $(0, 2)$-regular set S, then $|S| = \mathbf{j}^T \mathbf{g}_G(0, 2)$.*

Proof Supposing that $S \subset V(G)$ is a $(0, 2)$-regular set, according to Theorem 3, its characteristic vector x_S verifies

$$x_S = \mathbf{g}_G(0, 2) + \mathbf{q}.$$

Therefore,

$$|S| = \mathbf{j}^T x_S = \mathbf{j}^T \mathbf{g}_G(0, 2) + \mathbf{j}^T \mathbf{q}.$$

Since $\mathbf{q} = 0$ or $\mathbf{q} \in \mathscr{E}_G(\kappa - \tau)$ with $\kappa - \tau$ non-main, the conclusion follows. □

The following corollary provides a condition to decide when there are no $(0, 2)$-regular sets in G.

Corollary 1 *If $\mathbf{j}^T \mathbf{g}_G(0, 2)$ is not a natural number, then G has no $(0, 2)$-regular set.*

Now let us consider the particular case of graphs where $m(-2) = 0$.

Theorem 5 *If G is a graph such that $m(-2) = 0$, then G has a $(0, 2)$-regular set if and only if $\mathbf{g}_G(0, 2) \in \{0, 1\}^n$.*

Proof According to Theorem 3, since -2 is not an eigenvalue of G, there is a $(0, 2)$-regular set $S \subset V(G)$ if and only if $x_S = \mathbf{g}$. □

Considering a $m \times n$ matrix M and a vertex subset $I \subset V(G)$, M^I denotes the submatrix of M whose rows correspond to the indices in I.

Theorem 6 *Let G be a graph of order n such that $m(-2) > 0$ and let $\mathbf{U}$ be the $n \times m$ matrix whose columns are the eigenvectors of a basis of $\mathscr{E}_G(-2)$. If there is $v \in V(G)$ such that $\mathbf{U}^N$ (where $N = N_G[v] = \{v, v_1, \ldots, v_k\}$) has maximum rank, then it is possible to determine, in polynomial time, if G has a $(0, 2)$-regular set.*

Proof According to the necessary and sufficient condition for the existence of a (κ, τ)-regular set presented in Theorem 3, a vertex subset $S \subset V(G)$ is $(0, 2)$-regular if and only if its characteristic vector x_S is of the form

$$x_S = \mathbf{g}_G(0, 2) + \mathbf{q},$$

where $\mathbf{g}$ is defined by (3) and (4). Setting $\mathbf{q} = \mathbf{U}\beta$, where β is an m-tuple of scalars, such scalars may be determined solving the linear subsystem of $x_S = \mathbf{g} + \mathbf{q}$:

$$x_S^N = \mathbf{g}^N + \mathbf{U}^N\beta,$$

for each of the following possible instances of x_S :

$(x_S)_v = 1$ and then $(x_S)_{v_i} = 0, \forall i = 1, \ldots, k$;
$(x_S)_v = 0$ and then one of the following holds:

$(x_S)_{v_1} = (x_S)_{v_2} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_1, v_2\}$;
$(x_S)_{v_1} = (x_S)_{v_3} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_1, v_3\}$;
...
$(x_S)_{v_1} = (x_S)_{v_k} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_1, v_k\}$;
$(x_S)_{v_2} = (x_S)_{v_3} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_2, v_3\}$;
$(x_S)_{v_2} = (x_S)_{v_4} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_2, v_4\}$;
...
$(x_S)_{v_2} = (x_S)_{v_k} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_2, v_k\}$;
...
$(x_S)_{v_{k-1}} = (x_S)_{v_k} = 1$ and $(x_S)_{v_i} = 0, \forall v_i \in N_G[v] \setminus \{v_{k-1}, v_k\}$;

If for any of the cases described above the solution β is such that the obtained entries of vector x_S are $0-1$, then such x_S is the characteristic vector of a $(0,2)$-regular set. If none of the above instances generates a $0-1$ vector x_S, then we may conclude that the graph G has no $(0,2)$-regular set. Notice that each of the (at most) $1 + \binom{k}{2}$ linear systems under consideration can be solved in polynomial time, therefore it is possible to determine in polynomial time if G has a $(0,2)$-regular set. □

In order to generalize the procedure for the determination of $(0,2)$-regular sets to arbitrary graphs, it is worth to introduce some terminology. Let G be a graph with vertex set $V = \{1, \ldots, n\}$ and consider $I \subset V(G) = \{i_1, \ldots, i_m\}$. The m-tuple $x^I = (x_{i_1}, \ldots, x_{i_m}) \in \{0,1\}^m$ is $(0,2)$-*feasible* if it can be seen as a subvector of a characteristic vector $x \in \{0,1\}^n$ of a $(0,2)$-regular set in G. From this definition the following conditions hold:

(1) $\exists i_r \in I : x_{i_r} = 1 \Rightarrow \begin{cases} \forall i_j \in N_G(i_r) \cap I,\ x_{i_j} = 0 \quad \text{and} \\ \forall j \in N_G(i_r),\ \sum_{k \in N_G(j) \setminus \{i_r\}} x_k = 1; \end{cases}$
(2) $\exists i_s \in I : x_{i_s} = 0 \wedge N_G(i_s) \subseteq I \Rightarrow \sum_{j \in N_G[i_s]} x_j = 2.$

Using the $(0,2)$-feasible concept and consequently the above conditions, we are able to present an algorithm to determine a $(0,2)$-regular set in an arbitrary graph or to decide that no such set exists.

In the worst cases, steps 7–11 are executed 2^m times and, therefore, the execution of the algorithm is not polynomial. There is, however, a large number of graphs for which the described procedure is able to decide, in polynomial time, if there is a $(0,2)$-regular set and to determine it in the cases where it exists.

Algorithm 1 To determine a (0, 2)-regular set or decide that no such set exists

Input: (Graph G of order n, $m = m(-2)$ and matrix Q whose columns are the eigenvectors of a basis of $\mathscr{E}_G(-2)$).
Output: ((0, 2)-regular set of G or the conclusion that no such set exists).
1. **If** $\mathbf{j}^T g_G(0,2) \notin \mathbb{N}$ **then STOP** (there is no solution) **End If**;
2. **If** $m = 0$, **then STOP** ($x_S = g_G(0,2)$) **End If**;
3. **If** $\exists v \in V(G) : rank(Q^N) \leq d_G(v) + 1 \quad (N = N_G[v])$ **then STOP** (the output is a consequence of the low multiplicity results) **End If**;
4. Determine $I = \{i_1, \ldots, i_m\} \subset V(G) : rank(Q^I) = m$ and **set** $\mathbf{g} := \mathbf{g}_G(0,2)$;
5. **Set** *NoSolution* := *TRUE;*
6. **Set** $X := \{(x_{i_1}, \ldots, x_{i_m}) \text{ which is } (0,2)\text{-feasible for } G\}$;
7. **While** *NoSolution* $\wedge\, X \neq \emptyset$ **do**
8. Choose $(x_{i_1}, \ldots, x_{i_m}) \in X$ and **Set** $x^I := (x_{i_1}, \ldots, x_{i_m})^T$;
9. **Set** $X := X \backslash \{x^I\}$ and determine $\beta : x^I = \mathbf{g}^I + Q^I\beta$;
10. **If** $\mathbf{g} + Q\beta \in \{0,1\}^n$ **then** *NoSolution* := *FALSE* **End If**;
11. **End While**
12. **If** *NoSolution* = *FALSE* **then** $x := \mathbf{g} + Q\beta \in \{0,1\}^n$ **else** return *NoSolution*;
13. **End If.**
14. **End.**

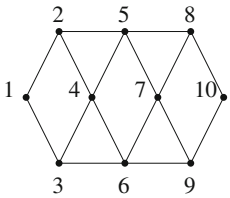


Fig. 1 Graph G

Example 1 Consider the graph G depicted in Fig. 1.

We will apply Algorithm 1 to determine a (0, 2)-regular set in G, a graph for which $m = m(-2) = 3$.

Since

$rank(\mathbf{j}) = 1,$

$rank(\mathbf{j}\ A_G\mathbf{j}) = 2,$

$rank(\mathbf{j}\ A_G\mathbf{j}\ (A_G)^2\mathbf{j}) = 2,$

we have $p = 2$ distinct main eigenvalues of G.

$$W = (\mathbf{j}\ A_G\mathbf{j}) = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 4 \\ 1 & 4 \\ 1 & 4 \\ 1 & 4 \\ 1 & 3 \\ 1 & 3 \\ 1 & 2 \end{pmatrix}.$$

The solution of the linear system $Wx = (A_G)^2\mathbf{j}$ is $C = \begin{pmatrix} -2 \\ 4 \end{pmatrix}$, so the coefficients of the main characteristic polynomial of G are $c_0 = 4, c_1 = -2$.

Next, the coefficients of vector $\mathbf{g} \in Main(G)$ will be determined.

Since

$$(\kappa - \tau)\alpha_0 = -\tau + \alpha_{p-1}c_{p-1}$$

and

$$(\kappa - \tau)\alpha_1 = \alpha_0 + \alpha_{p-1}c_{p-2},$$

we have $\alpha_0 = \frac{6}{7}, \alpha_1 = -\frac{1}{7}$, hence

$$\mathbf{g} = \alpha_0\mathbf{j} + \alpha_1 A_G\mathbf{j} = \begin{pmatrix} 0.2857 \\ 0.4286 \\ 0.2857 \\ 0.2857 \\ 0.4286 \\ 0.5714 \\ 0.4286 \\ 0.2857 \\ 0.4286 \\ 0.5714 \end{pmatrix}.$$

Considering matrix Q whose columns q_1, q_2, q_3 form a basis for the eigenspace associated to eigenvalue -2, we will proceed, searching for a vertex v for which the submatrix of Q corresponding to $N_G[v]$ has full rank.

$$Q = \begin{pmatrix} 0.4962 & 0.0345 & -0.1431 \\ -0.4962 & -0.0345 & 0.1431 \\ -0.4962 & -0.0345 & 0.1431 \\ 0.2335 & -0.1035 & -0.5878 \\ 0.2626 & 0.1380 & 0.4446 \\ 0.2626 & 0.1380 & 0.4446 \\ -0.2626 & 0.3795 & -0.4446 \\ 0 & -0.5175 & 0 \\ 0 & -0.5175 & 0 \\ 0 & 0.5175 & 0 \end{pmatrix}.$$

The obtained results are summarized in the following table.

v	$N_G(v)$	Rank$(Q^{N_G[v]})$
1	2, 3	1
2	1, 4, 5	2
3	1, 4, 6	2
4	2, 3, 5, 6	2
5	2, 4, 7, 8	3

.

It is obvious that the submatrix of Q corresponding to lines $2, 4, 5, 7$ and 8, the closed neighbourhood of vertex 5, has full rank, so we will consider the subvector of $\mathbf{g}$ and the submatrix of Q corresponding to $I = N_G[5] = \{2, 4, 5, 7, 8\}$.

$$\mathbf{g}^I = \begin{pmatrix} 0.4286 \\ 0.2857 \\ 0.2857 \\ 0.2857 \\ 0.4286 \end{pmatrix}, Q^I = \begin{pmatrix} -0.4962 & -0.0345 & 0.1431 \\ 0.2335 & -0.1035 & -0.5878 \\ 0.2626 & 0.1380 & 0.4446 \\ -0.2626 & 0.3795 & -0.4446 \\ 0 & -0.5175 & 0 \end{pmatrix}.$$

Supposing that G has a $(0, 2)$-regular set S, there are two possibilities to be considered: whether $5 \in S$ or $5 \notin S$ and there are $1+\binom{4}{2}$ possible instances $(1) - (7)$ for the entries of x_S that correspond to $N_G[5]$ (see next table).

Inst.	$(x_S)_1$	$(x_S)_2$	$(x_S)_3$	$(x_S)_4$	$(x_S)_5$	$(x_S)_6$	$(x_S)_7$	$(x_S)_8$	$(x_S)_9$	$(x_S)_{10}$
(1)	$*$	0	$*$	0	1	$*$	0	0	$*$	$*$
(2)	$*$	1	$*$	1	0	$*$	0	0	$*$	$*$
(3)	$*$	1	$*$	0	0	$*$	1	0	$*$	$*$
(4)	$*$	1	$*$	0	0	$*$	0	1	$*$	$*$
(5)	$*$	0	$*$	1	0	$*$	1	0	$*$	$*$
(6)	$*$	0	$*$	1	0	$*$	0	1	$*$	$*$
(7)	$*$	0	$*$	0	0	$*$	1	1	$*$	$*$

.

Supposing that $5 \in S$, the entries of x_S corresponding to $I = \{2, 4, 5, 7, 8\}$ are

$$x_S^I = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

and the solution of the subsystem

$$x_S^I = g^I + \beta_1 (q^I)_1 + \beta_2 (q^I)_2 + \beta_3 (q^I)_3$$

is $\beta_1 = 1.0215, \beta_2 = 0.8281, \beta_3 = 0.7461$.

Solving the complete system and calculating $\mathbf{g} + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3$ for the evaluated values of β_1, β_2 and β_3, the following result is obtained

$$x_S = \mathbf{g} + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

and $S = \{1, 5, 6, 10\}$ is a $(0, 2)$-regular set of G.

4 Application: Determination of Maximum Matchings

In this section, Algorithm 1 is combined with the procedure for maximum matchings described in [1], to provide a strategy for the determination of maximum matchings in arbitrary graphs. Such strategy is based on the determination of $(0, 2)$-regular sets in the correspondent line graphs, in the cases where they occur, or on the addition of extra vertices to the original graphs, in the situations where the line graphs under consideration have no $(0, 2)$-regular sets.

Considering the graph described in Example 1 and the $(0, 2)$-regular set determined by Algorithm 1, it is easily checkable that it corresponds to a maximum matching in graph G whose line graph is $L(G)$. Both graphs are depicted in Fig. 2.

It should be noticed that, according to Theorem 7 in [1], a graph G which is not a star neither a triangle has a perfect matching if and only if its line graph has a $(0, 2)$-regular set.

We will now determine a maximum matching in a graph whose line graph does not have a$(0, 2)$-regular set, which is equivalent to say that the root graph has no perfect matchings, following the algorithmic strategy proposed in [1].

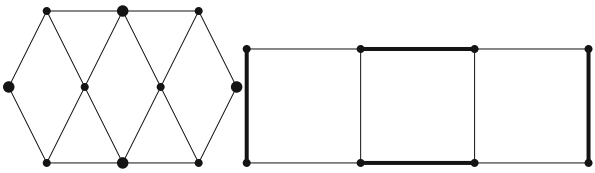
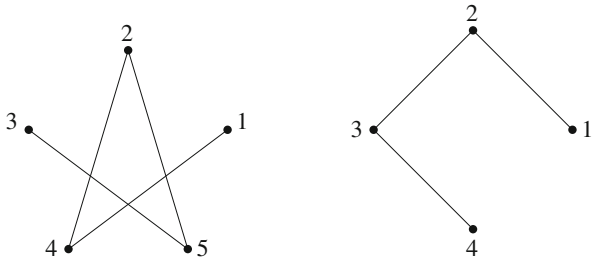


Fig. 2 Graphs $L(G)$ and G

Fig. 3 Graphs G_1 and $L(G_1)$



Example 2 Consider graphs G_1 and $L(G_1)$ both depicted in Fig. 3.

Since -2 is not an eigenvalue of $L(G_1)$, we will determine the parametric vector $\mathbf{g}_{L(G_1)}(0,2)$ in order to find out if its coordinates are $0-1$. $L(G_1)$ has two main eigenvalues and its walk matrix is

$$W = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 2 \\ 1 & 1 \end{pmatrix}.$$

The coefficients of the main characteristic polynomial of $L(G_1)$, that are the solutions of system $Wx = A_{L(G_1)}\mathbf{j}$, are $c_0 = c_1 = 1$. The corresponding solutions of system (4), that is, the coefficients of $\mathbf{g}_{L(G_1)}(0,2)$, are $\alpha_0 = \frac{6}{5}$ and $\alpha_1 = -\frac{2}{5}$. Therefore,

$$\mathbf{g}_{L(G_1)}(0,2) = \alpha_0\mathbf{j} + \alpha_1 A_{L(G_1)}\mathbf{j} = \begin{pmatrix} 0.8 \\ 0.4 \\ 0.4 \\ 0.8 \end{pmatrix}$$

and it can be concluded that the graph $L(G_1)$ has no $(0,2)$-regular sets. In order to determine a maximum matching in G_1, we will proceed as it is proposed in [1]. Since G_1 has an odd number of vertices, a single vertex will be added to G_1 and connected to all its vertices. The graph G_2 and its line graph $L(G_2)$, depicted in Fig. 4, are obtained.

Repeating the procedure described in Algorithm 1 (now applied to $L(G_2)$), we have that $m(-2) = 3$ and $p = 4$. It is also easy to verify that $c_0 = 5, c_1 = 1, c_2 = -6$ and $c_3 = 0$ are the coefficients of the main characteristic polynomial of $L(G_2)$. The solution of system (4) is $\alpha_0 = 1, \alpha_1 = -\frac{13}{20}, \alpha_2 = \frac{7}{20}, \alpha_3 = -\frac{1}{20}$ and the corresponding parametric vector $\mathbf{g}_{L(G_2)}(0,2)$ is

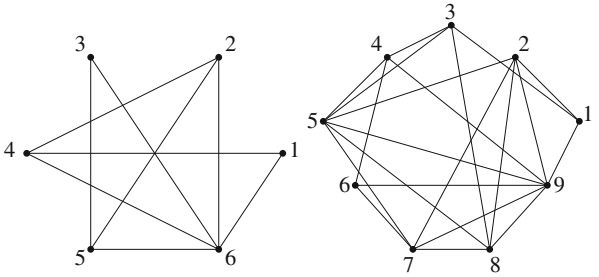


Fig. 4 Graphs G_2 and $L(G_2)$

$$\mathbf{g}_{L(G_2)}(0,2) = \alpha_0\mathbf{j} + \alpha_1 A_{L(G_2)}\mathbf{j} + \alpha_2 (A_{L(G_2)})^2\mathbf{j} + \alpha_3 (A_{L(G_2)})^3\mathbf{j} = \begin{pmatrix} 0.6 \\ 0.4 \\ 0.4 \\ 0.4 \\ 0.2 \\ 0.6 \\ 0.4 \\ 0 \\ 0 \end{pmatrix}.$$

We will now consider matrix Q, whose columns form a basis for the eigenspace associated to the eigenvalue -2 of $L(G_2)$.

$$Q = \begin{pmatrix} 0.2241 & 0.4458 & 0.2260 \\ -0.2241 & -0.4458 & -0.2260 \\ 0.3176 & -0.4525 & -0.3072 \\ -0.4115 & 0.3651 & -0.3121 \\ 0.0939 & 0.0875 & 0.6193 \\ -0.1302 & -0.3583 & 0.3933 \\ 0.1302 & 0.3583 & -0.3933 \\ -0.5416 & 0.0068 & 0.0812 \\ 0.5416 & -0.0068 & -0.0812 \end{pmatrix}.$$

Searching for a vertex of degree ≥ 2 in $L(G_2)$ whose closed neighbourhood corresponds to a submatrix of Q with maximum rank, the following table is obtained.

v	$N_{L(G_2)}(v)$	$\text{rank}(Q^{N_{L(G_2)}[v]})$
1	2, 3, 8	2
2	1, 5, 7, 8, 9	3
3	1, 4, 5, 8	3
4	3, 5, 6, 9	3
5	2, 3, 4, 7, 8, 9	3
6	4, 7, 9	2
7	2, 5, 6, 8, 9	3
8	1, 2, 3, 5, 7, 9	3
9	2, 4, 5, 6, 7, 8	3

.

It is evident that the closed neighbourhood of vertex 2 verifies the mentioned requirements and we will proceed considering the subvector of $\mathbf{g}$ and the submatrix of Q whose lines are the elements of $N_{L(G_2)}[2]$.

$$\mathbf{g}^I = \begin{pmatrix} 0.6 \\ 0.4 \\ 0.2 \\ 0.4 \\ 0 \\ 0 \end{pmatrix}, Q^I = \begin{pmatrix} 0.2241 & 0.4458 & 0.2260 \\ -0.2241 & -0.4458 & -0.2260 \\ 0.0939 & 0.0875 & 0.6193 \\ 0.1302 & 0.3583 & -0.3933 \\ -0.5416 & 0.0068 & 0.0812 \\ 0.5416 & -0.0068 & -0.0812 \end{pmatrix}.$$

Supposing that $L(G_2)$ contains a $(0, 2)$-regular set, there are $1+\binom{5}{2}$ possible instances for the entries of x_S that correspond to $N_{G_2}[2]$. One of them is

$(x_S)_1$	$(x_S)_2$	$(x_S)_3$	$(x_S)_4$	$(x_S)_5$	$(x_S)_6$	$(x_S)_7$	$(x_S)_8$	$(x_S)_9$
0	1	$*$	$*$	0	$*$	0	0	0

.

Assuming that $2 \in S$, the entries $1, 2, 5, 7, 8, 9$ of x_S must be of the form

$$x_S^I = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and the solution of the subsystem

$$x_S^I = \mathbf{g}^I + \beta_1 (q^I)_1 + \beta_2 (q^I)_2 + \beta_3 (q^I)_3$$

is $\beta_1 = -0.0367, \beta_2 = -1.2566, \beta_3 = -0.1399$.

Solving the complete system $x_S = \mathbf{g} + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3$ and computing $\mathbf{g} + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3$, we obtain

$$x_S = \mathbf{g} + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and $S = \{2, 3, 6\}$ is a $(0, 2)$-regular set of $L(G_2)$. The resulting $(0, 2)$-regular set S corresponds to the perfect matching of G_2: $M^* = \{\{1, 4\}, \{2, 5\}, \{3, 6\}\}$. Therefore, intersecting the edges of M^* with the edge set of the root graph G_1, a maximum matching of G_1

$$M = \{\{1, 4\}, \{2, 5\}\},$$

is determined.

5 Final Remarks

The aim of this paper is the introduction of an algorithm for the determination of $(0, 2)$-regular sets in arbitrary graphs. In Sect. 2, an overview of the most relevant results about the determination of (κ, τ)-regular sets using the main eigenspace of a given graph is presented. Such results were introduced [3]. In Sect. 3, several results that lead to the determination of $(0, 2)$-regular sets are introduced and a new algorithm that determines a $(0, 2)$-regular set in an arbitrary graph or concludes that no such set exists is also described. Section 4 is devoted to the application of the introduced algorithm to the determination of maximum matchings.

Despite the interest of the introduced techniques for the determination of $(0, 2)$-regular sets in general graphs, their particular application to the determination of maximum matchings is not efficient in many cases. The use of these techniques in this context is for illustrating the application of the algorithm. It remains as an open problem, to obtain additional results for improving the determination of $(0, 2)$-regular sets in line graphs.

References

1. Cardoso, D.M.: Convex quadratic programming approach to the maximum matching problem. J. Glob. Optim. **21**, 91–106 (2001)
2. Cardoso, D.M., Rama, P.: Equitable bipartitions of graphs and related results. J. Math. Sci. **120**, 869–880 (2004)
3. Cardoso, D.M., Sciriha, I., Zerafa, C.: Main eigenvalues and (κ, τ)-regular sets. Lin. Algebra Appl. **432**, 2399–2408 (2010)
4. Cvetković, D., Doob, M., Sachs, H.: Spectra of Graphs. Academic, New York (1979)
5. Rowlinson, P.: The main eigenvalues of a graph: a survey. Appl. Anal. Discr. Math. **1**, 445–471 (2007)

A Multiobjective Electromagnetism-Like Algorithm with Improved Local Search

Pedro Carrasqueira, Maria João Alves, and Carlos Henggeler Antunes

Abstract The Multiobjective Electromagnetism-like Mechanism (MOEM) is a relatively new technique for solving continuous multiobjective optimization problems. In this work, an enhanced MOEM algorithm (EMOEM) with a modified local search phase is presented. This algorithm derives from the modification of some key components of MOEM including a novel local search strategy, which are relevant for improving its performance. To assess the new EMOEM algorithm, a comparison with an original MOEM algorithm and other three multiobjective optimization state-of-the-art approaches, OMOPSO (a multiobjective particle swarm optimization algorithm), MOSADE (a multiobjective differential evolution algorithm) and NSGA-II (a multiobjective evolutionary algorithm), is presented. Our aim is to assess the ability of these algorithms to solve continuous problems including benchmark problems and an inventory control problem. Experiments show that EMOEM performs better in terms of convergence and diversity when compared with the original MOEM algorithm. EMOEM is also competitive in comparison with the other state-of-art algorithms.

1 Introduction

In the last two decades several meta-heuristics have been developed to address multiobjective optimization problems. The importance of multiobjective models in practical applications and the difficulties that arise in their resolution have fostered research in this field. In general, these problems have an extensive

P. Carrasqueira (✉)
INESC Coimbra, Coimbra, Portugal
e-mail: pmcarrasqueira@net.sapo.pt

M.J. Alves
Faculty of Economics, University of Coimbra/INESC Coimbra, Coimbra, Portugal
e-mail: mjalves@fe.uc.pt

C.H. Antunes
Department of Electrical and Computer Engineering, University of Coimbra/INESC Coimbra, Coimbra, Portugal
e-mail: ch@deec.uc.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_8

set of non-dominated solutions, which may be difficult to compute. In addition to multiobjective evolutionary algorithms (MOEA), such as NSGA-II [5], other population-based meta-heuristics as particle swarm optimization (MOPSO), differential evolution (MODE) and electromagnetism-like mechanism (MOEM) have also been proposed. These approaches were initially developed to solve single objective optimization problems and then adapted to multiobjective optimization.

Particle Swarm Optimization (PSO) is inspired on the behavior of some species, such as bird flocks when they are looking for food [10]. Population members move themselves based on their own experience and the experience of their neighbors. Several researchers have proposed modifications to the initial algorithm and later this approach was extended to solve multiobjective optimization problems (MOPSO). Most of these approaches rely on the Pareto Dominance concept. Among these, OMOPSO [15] has been considered one of the most competitive MOPSO algorithms [7].

In Differential Evolution (DE) [16] population members evolve through a mechanism based on solution vector differences aimed at capturing fitness landscape. Despite its simplicity, this mechanism proved to be very effective [14]. The adaptation of the algorithm to solve multiobjective optimization problems only requires a slight modification of the selection mechanism. In the last decade, several versions of multiobjective DE approaches (MODE) have been developed. Some of them solve the problem as a single objective problem by encompassing the multiple objective functions into a scalar function, but most MODE algorithms are Pareto-based approaches. In addition, some of these algorithms adopt mechanisms of multiobjective evolutionary algorithms, namely non-dominated sorting [5] and diversity preserving techniques based on the crowding measure. In [11], a review of the state-of-the-art of MODE algorithms is presented. MOSADE [22] is a recently developed algorithm that incorporates self-adaptation of the parameters and a crowding entropy strategy, which is able to measure the crowding degree of the solutions more accurately.

The Electromagnetism-like mechanism (EM) is a recent meta-heuristic introduced in [3] and research has been conducted concerning EM applications to single objective continuous optimization problems. This approach is inspired by the attraction-repulsion mechanism of the electromagnetism theory. After initialization, the EM algorithm is composed by three main steps: local search procedure, individual force vector evaluation and population individuals' movement. Concerning single objective optimization, several different designs of these components have been proposed [13, 25]. The performance of the EM algorithm is strongly dependent on these components. The individual force vector evaluation and the movement of the population individuals are influenced by all the other population members. In contrast, applying local search procedure to a selected individual does not use information about the other individuals. The local search procedure is therefore a decisive component of the performance of an EM algorithm.

EM was adapted to solve multiobjective problems by Tsou and Kao [19]. This algorithm, which we will denote hereafter by MOEM, was also used to solve an inventory control problem in [20, 21]. As in the single objective case, the

MOEM algorithm relies on the individual force vector evaluation, the population individuals' movement and the local search process. Aiming to improve MOEM performance, we have developed an Enhanced MOEM algorithm (EMOEM) [4] in which the individual force vector and the position updating process were modified in order to overcome some difficulties experienced by the MOEM algorithm. In the present work, we propose a new version of the EMOEM algorithm, which incorporates an improved local search strategy. This strategy uses the Hooke and Jeeves's algorithm [9], which has been successfully applied to single objective optimization [17] and to multiobjective optimization [12]. The new EMOEM algorithm is compared with MOEM [19], OMOPSO [15], MOSADE [22] and NSGA-II [5] algorithms. The results obtained confirm that EMOEM outperforms the MOEM algorithm. Also, it is very competitive with respect to the other state-of-the-art algorithms.

The remainder of the paper is organized as follows. In Sect. 2 the formulation of a continuous multiobjective optimization problem is presented. In Sect. 3, the OMOPSO, MOSADE and NSGA-II algorithms are briefly described. In Sect. 4 the new EMOEM algorithm is presented. In Sect. 5, results are analyzed. Section 6 provides some conclusions and future work directions.

2 Multiobjective Optimization

A multiobjective optimization problem is defined as

$$\begin{aligned} &Min\, \overrightarrow{f}(\overrightarrow{x}) = (f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), \cdots, f_m(\overrightarrow{x})) \\ &S.t.\, \overrightarrow{x} \in \Omega, \\ &\Omega = \{\overrightarrow{x} \in \mathbb{R}^d | g_i(\overrightarrow{x}) \leq 0,\ i = 1, 2, \cdots, p\}, \end{aligned} \tag{1}$$

where $\overrightarrow{f}(\overrightarrow{x})$ is the vector of objective functions to optimize, $\overrightarrow{x} = (x_1, x_2, \cdots, x_d)$ is the decision vector, d is the number of variables and $g_i(\overrightarrow{x}) \leq 0, i = 1, 2, \cdots, p$ are the constraints.

Definition 1 A vector $\overrightarrow{x} \in \Omega$ dominates a vector $\overrightarrow{y} \in \Omega$ and we say $\overrightarrow{x} \prec \overrightarrow{y}$, if $f_i(\overrightarrow{x}) \leq f_i(\overrightarrow{y}),\ \forall i = 1, \cdots, m$ and $\exists j \in \{1, \cdots, m\} : f_j(\overrightarrow{x}) < f_j(\overrightarrow{y})$.

Definition 2 A solution $\overrightarrow{x} \in \Omega$ is said efficient if $\nexists \overrightarrow{y} \in \Omega : \overrightarrow{y} \prec \overrightarrow{x}$. The corresponding objective point $\overrightarrow{f}(\overrightarrow{x})$ is a non-dominated point.

These definitions are valid for minimization problems. Similar definitions can be derived for maximization problems. The set of all non-dominated solutions to a multiobjective optimization problem is called the Pareto optimal front. Our goal is to approximate the entire Pareto optimal front of the problem.

3 Description of State-of-the-Art MOEA, MOPSO and MODE Algorithms

3.1 NSGA-II Algorithm

NSGA-II [5] is a well-known multiobjective evolutionary algorithm representative of the state-of-the-art of its class. The algorithm uses the genetic operators crossover, mutation and selection to generate a new population. A non-dominated sorting mechanism is introduced to rank population members by non-dominated fronts. Also a crowding distance operator is applied to each population member to sort individuals within the fronts by decreasing order of the crowding distance. At each generation the algorithm starts by creating the new population of n individuals using the genetic operators. Then, the old and new populations are joined. The resultant population is ranked and sorted by non-dominated fronts using a fast sorting mechanism. The individuals in the same front are sorted by decreasing order of their crowding distance. The best n individuals are selected to create the new population for the next generation. This process continues until the stop criterion is reached, and the final population is presented.

3.2 OMOPSO Algorithm

The PSO algorithm is usually initialized with a population of n randomly generated particles. Each particle is assigned a "velocity" operator that indicates the direction and amplitude of the particle movement resulting from the combination of the directions of the best position so far achieved by the particle ($pbest_i$) and the best position attained by the whole population ($gbest$). In multiobjective optimization, $pbest_i$ and $gbest$ are not unique. Each particle $\overrightarrow{x}_i$ moves itself in the k iteration according to the expressions

$$\overrightarrow{v}_i^{k+1} = w\overrightarrow{v}_i^{k} + c_1 rand()(pbest_i - \overrightarrow{x}_i^{k}) + c_2 rand()(gbest - \overrightarrow{x}_i^{k}) \quad (2)$$

$$\overrightarrow{x}_i^{k+1} = \overrightarrow{x}_i^{k} + \overrightarrow{v}_i^{k+1}, i = 1, 2, \cdots, n \quad (3)$$

where w, c_1 and c_2 are parameters of the algorithm, $rand(\ )$ is a random uniform value in the interval $[0, 1]$ and n is the number of particles in the population. The way that parameters w, c_1 and c_2 are defined during the execution of the algorithm depends on the MOPSO version. In case of OMOPSO [15], these parameters are randomly chosen, in each iteration, within a predefined interval: $w \in [0.1, 0.5]$ and $c_1, c_2 \in [1.5, 2]$. Other features of OMOPSO are the following, some of them being shared with other MOPSO algorithms.

- A new particle i replaces its $pbest_i$ if $pbest_i$ is dominated by the new particle or both are non-dominated with respect to each other.

- Non-dominated particles are stored in an external archive. This archive has a predefined size. When the capacity of the archive is reached, one particle is inserted into the archive after deleting a particle from the archive.
- A crowding distance [5] is assigned to each particle of the external archive. This measure is used to select the leader ($gbest_i$) of each population member and to select a particle of the external archive to be replaced when the archive is full.
- A mutation operator named turbulence is used, with a certain probability. This operation is performed after the execution of operations (2) and (3). The population is split in three parts. One third is applied a uniform mutation, another third is applied a non-uniform mutation and the last third of the population is not changed.
- An additional archive is used in [15]. This archive is the output of the algorithm, and results from applying the concept of $\epsilon - dominance$ to the external archive of non-dominated particles. This mechanism limits the size of the non-dominated archive. In further implementations of OMOPSO algorithm, this additional archive has not been considered [7]. We also do not use this archive.

The OMOPSO algorithm has been compared with other MOPSO and MOEA algorithms in [7] and due to its good performance OMOPSO has become one of the representatives of the state-of-the-art MOPSO approaches. The pseudo code of the OMOPSO algorithm is presented in Algorithm 1.

Algorithm 1 OMOPSO pseudo code

Initialize iteration counter, $k = 1$
Randomly initialize each population individual $\overrightarrow{x}_i^k$, and its velocity $\overrightarrow{v}_i^k, i = 1, \cdots, n$
Assess each particle $\overrightarrow{x}_i^k$, evaluating $\overrightarrow{f}(\overrightarrow{x}_i^k) = (f_1(\overrightarrow{x}_i^k), f_2(\overrightarrow{x}_i^k), \cdots, f_m(\overrightarrow{x}_i^k)), \quad i = 1, \cdots, n$
Initialize $pbest_i$ as the individual particle $\overrightarrow{x}_i^k, i = 1, \cdots, n$
Insert non-dominated particles into external archive
Sort archive members by decreasing order of crowding distance
while stop criterion is not met **do**
    for $i = 1, \cdots, n$ **do**
        Select the leader $gbest_i$
        Update particle velocity $\overrightarrow{v}_i^{k+1}$, using (2)
        Update particle position $\overrightarrow{x}_i^{k+1}$, using (3)
        Mutate particle $\overrightarrow{x}_i^{k+1}$
        Assess the particle $\overrightarrow{x}_i^{k+1}$, evaluating $f(\overrightarrow{x}_i^{k+1})$
        Update $pbest_i$ of particle i
    end for
    Update external archive of non-dominated solutions
    Sort archive members by decreasing order of crowding distance
end while
Return archive of non-dominated solutions

3.3 MOSADE Algorithm

In Differential Evolution (DE) a random generated population evolves based on three mechanisms: mutation, recombination and selection. There are a number of different ways to define such operations. We will consider the parameterization DE/rand/1/bin. It means that each basis solution is randomly chosen, the mutation operator uses only one vector and the binomial distribution is applied in the crossover operation.

At each generation k and for each solution i, the algorithm starts by applying a mutation operator. To perform this operation, three individuals of the population are randomly chosen, say r_1, r_2 and r_3. A new solution is built, which is called the donor vector:

$$\overrightarrow{v}_i^{k+1} = \overrightarrow{x}_{r_1}^k + F(\overrightarrow{x}_{r_2}^k - \overrightarrow{x}_{r_3}^k) \tag{4}$$

where $F > 0$ is the mutation parameter. This parameter controls the extent of the movement performed. Low values of F favor exploitation and high values favor exploration. Then, a recombination operation is applied. The trial vector $\overrightarrow{u}_i^{k+1}$ is developed based on the elements of the vector $\overrightarrow{x}_i^k$ and the donor vector $\overrightarrow{v}_i^{k+1}$:

$$u_{ij}^{k+1} = \begin{cases} v_{ij}^{k+1} & \text{if } rand(j) \leq CR \text{ or } j = randint(i) \\ x_{ij}^k & \text{if } rand(j) > CR \text{ and } j \neq randint(i) \end{cases}, j = 1, \cdots, d \tag{5}$$

where $rand(j) \sim U[0, 1]$, CR is the recombination probability and $randint$ is a random integer between 1 and d. To decide which solution will survive to the next generation, a selection operation is performed.

In MOSADE algorithm [22], the selection operation considers the Pareto dominance between two individuals and, in case they are non-dominated with respect to each other, the least crowded solution is selected. This operation is given by the expression

$$\overrightarrow{x}_i^{k+1} = \begin{cases} \overrightarrow{x}_i^k & \text{if } \overrightarrow{x}_i^k \prec \overrightarrow{u}_i^{k+1} \\ \overrightarrow{u}_i^{k+1} & \text{if } \overrightarrow{u}_i^{k+1} \prec \overrightarrow{x}_i^k \\ LC(\overrightarrow{u}_i^{k+1}, \overrightarrow{x}_i^k) & \text{if } \overrightarrow{u}_i^{k+1} \prec\succ \overrightarrow{x}_i^k \end{cases} \tag{6}$$

where $LC(\overrightarrow{u}_i^{k+1}, \overrightarrow{x}_i^k)$ means the less crowded individual $\overrightarrow{x}_i^k$ or $\overrightarrow{u}_i^{k+1}$, if they do not dominate each other ($\prec\succ$).

In MOSADE, non-dominated particles found are stored in an external archive. The solutions in this archive are distinguished in terms of a diversity preserving mechanism. The authors [22] introduce a new mechanism called crowding-entropy operator based on the crowding distance proposed by Deb et al. [5] that aims at a better diversity of solutions in the non-dominated external archive. The entropy concept is employed to describe the distribution of a solution along each one of

Algorithm 2 MOSADE pseudo code

1: Initialize iteration counter, $k = 1$
2: Random initialize each population individual $\overrightarrow{x}_i^k,\ i = 1, \cdots, n$
3: Assess each particle $\overrightarrow{x}_i^k$, evaluating $\overrightarrow{f}(\overrightarrow{x}_i^k) = (f_1(\overrightarrow{x}_i^k), f_2(\overrightarrow{x}_i^k), \cdots, f_m(\overrightarrow{x}_i^k)),\ i = 1, \cdots, n$
4: Initialize external archive of non-dominated solutions
5: **while** stop criterion is not met **do**
6: **for** $i = 1, \cdots, n$ **do**
7: Randomly select r_1, r_2 and $r_3 \in \{1, 2, \cdots, n\}$
8: Create the donor vector $\overrightarrow{v}_i^{k+1}$ by mutation using (4)
9: Create the vector $\overrightarrow{u}_i^{k+1}$ by recombination of $\overrightarrow{x}_i^k$ and $\overrightarrow{v}_i^{k+1}$ using (5)
10: Assess $\overrightarrow{u}_i^{k+1}$ evaluating $\overrightarrow{f}(\overrightarrow{u}_i^{k+1})$
11: Perform selection between $\overrightarrow{x}_i^k$ and $\overrightarrow{u}_i^{k+1}$ to define $\overrightarrow{x}_i^{k+1}$, using (6)
12: **end for**
13: Update external archive of non-dominated solutions
14: **end while**
15: Return archive of non-dominated solutions

the objectives. If a point is located in the middle of its neighbors, it has a better distribution than a point with the same average distance but which is near one neighbor and far from another. The parameters F and CR are set independently for each individual and they are self-adaptive. This means that if one solution has not been improved during a certain number of generations, the F and CR parameters assigned to that solution are recalculated in a random manner within a predefined range. The authors argue that the DE algorithm is very sensitive to parameter values and with this strategy the parameters do not need to be fine tuned. The MOSADE algorithm was also designed to solve constrained problems. The mechanism to deal with constraints presented in [5] is adopted in MOSADE. This mechanism is based on the non-dominance concept and the total amount of constraint violation, for each individual. This approach was tested on benchmark problems and it obtained competitive results. The pseudo code of the MOSADE algorithm is presented in Algorithm 2.

4 Multiobjective Electromagnetism-Like Mechanism Algorithms

4.1 MOEM Algorithm

The first attempt to design a MOEM algorithm was presented in [19]. The algorithm is based on three main components: individual charge, total force, and local search procedure. The algorithm starts with a randomly generated population of individuals. Then, the population evolves by local search and population movement

based on the computation of individual charges and attraction/repulsion forces, until a predefined criterion is met.

The local search procedure is applied to each particle of a local archive $\tilde{S}$. Each variable of the particle is changed by a random value. When all the variables have been changed the particle is evaluated. If the new generated particle dominates the old particle, this is replaced. The process is repeated *lsiter* times for each particle in each generation.

The charge of an individual depends on its objective function values and the ones of all other population members. In MOEM, the charge of a particle i is given by

$$q_i = exp(-d\frac{\min_{\overrightarrow{x_p}\in\tilde{S}} \|\overrightarrow{f}(\overrightarrow{x}_i) - \overrightarrow{f}(\overrightarrow{x}_p)\|}{\sum_{j=1}^{n} \min_{\overrightarrow{x_p}\in\tilde{S}} \|\overrightarrow{f}(\overrightarrow{x}_j) - \overrightarrow{f}(\overrightarrow{x}_p)\|}),\ i = 1, \cdots, n \tag{7}$$

where $\tilde{S}$ is the local archive, which is a subset of the external archive that stores the non-dominated solutions found by the algorithm.

In the Coulomb's Law, the force exerted between two particles is inversely proportional to the square of their distance and directly proportional to the product of their charges. The individual force that each $\overrightarrow{x}_j$ exerts on another $\overrightarrow{x}_i$ resembles this principle as it is given by (8):

$$\overrightarrow{F}^{ij} = \begin{cases} (\overrightarrow{x}_i - \overrightarrow{x}_j)\frac{q_i q_j}{\|\overrightarrow{x}_i - \overrightarrow{x}_j\|^2} & if\ \overrightarrow{x}_i \prec \overrightarrow{x}_j \\ (\overrightarrow{x}_j - \overrightarrow{x}_i)\frac{q_i q_j}{\|\overrightarrow{x}_i - \overrightarrow{x}_j\|^2} & otherwise \end{cases},\ (j \neq i) \tag{8}$$

The total force exerted on individual i is the sum of individual forces:

$$\overrightarrow{F}^{i} = \sum_{j\neq i}^{n} \overrightarrow{F}^{ij} \tag{9}$$

After obtaining the total force vector the movement of each individual i is performed according to expression (10)

$$x_{ir}^{k+1} = \begin{cases} x_{ir}^{k} + \lambda\frac{F_r^i}{\|\overrightarrow{F}^i\|}(u_r - x_{ir}^{k})\ if\ F_r^i > 0 \\ x_{ir}^{k} + \lambda\frac{F_r^i}{\|\overrightarrow{F}^i\|}(x_{ir}^{k} - l_r)\ if\ F_r^i \leq 0 \end{cases},\ r = 1, \cdots, d \tag{10}$$

where λ is a random number such that $\lambda \sim U(0, 1)$ and l_r, u_r are the lower and upper bounds for each component r of particle $\overrightarrow{x}_i$, respectively. F_r^i is the $r - component$ of the total force exerted on individual i.

The non-dominated particles obtained during the algorithm execution are stored in the external archive. In [19], the clustering technique proposed in [23] was used to maintain the diversity of the non-dominated archive.

The main steps of the MOEM algorithm are described in Algorithms 3 and 4 details the local search procedure.

Algorithm 3 MOEM pseudo code

Initialize iteration counter, $k = 1$
Randomly initialize each population individual $\overrightarrow{x}_i^k,\ i = 1, \cdots, n$
Assess each particle $\overrightarrow{x}_i^k$ evaluating $\overrightarrow{f}(\overrightarrow{x}_i^k) = (f_1(\overrightarrow{x}_i^k), f_2(\overrightarrow{x}_i^k), \cdots, f_m(\overrightarrow{x}_i^k)),\ i = 1, \cdots, n$
Insert non-dominated particles into archive $\tilde{A}$
while stop criterion is not met **do**
    Randomly select the particles to insert into the local archive $\tilde{S}$ from archive $\tilde{A}$
    Do local search in $\tilde{S}$
    for $i = 1, \cdots, n$ **do**
        Compute the charge (q_i) of the particle i using (7)
        Compute total force ($\overrightarrow{F}^i$) exerted on particle i using (8) and (9)
        Move particle i using (10)
        Update non-dominated archive $\tilde{A}$
    end for
end while
Return non-dominated archive $\tilde{A}$

Algorithm 4 Local Search (MOEM) pseudo code

$length = \delta \cdot (\max_{k=1,\cdots,d}\{u_k - l_k\})$
for $i = 1, \cdots, |\tilde{S}|$ **do**
    Initialize local search counter, $count = 0$
    while $count < lsiter$ **do**
        $\overrightarrow{z} = \overrightarrow{x}_i$
        for $k = 1, \cdots, d$ **do**
            Select two random values $\lambda_1, \lambda_2 \in [0, 1]$
            if $\lambda_1 > 0.5$ **then**
                $z_k = z_k + \lambda_2 \times length$
            else
                $z_k = z_k - \lambda_2 \times length$
            end if
        end for
        if $\overrightarrow{z} \prec \overrightarrow{x}_i$ **then**
            Update $\tilde{A}$ with $\overrightarrow{z}$
            $\overrightarrow{x}_i = \overrightarrow{z}$
        end if
        $count = count + 1$
    end while
end for

4.2 EMOEM Algorithm

The EMOEM algorithm proposed in [4] modified some of the main components of the MOEM algorithm. In MOEM, the movement of each particle (10) is influenced by its force vector (8–9), which depends on the charges of all the other particles of the population (7). As can be observed in (7), the charge is computed using the number of variables (d) as a correcting factor, in order to emphasize the differences among individuals. However, the differences of charge values between particles performing very different in the objective space could vanish when the ratio of the number of variables to the population size is small. In the enhanced algorithm EMOEM, the number of variables is replaced by the population size in the factor used for the charge computation. Thus, in EMOEM the charge is obtained using the expression (11).

$$q_i = exp(-n\frac{\min_{\overrightarrow{x_p}\in\tilde{S}} \|\overrightarrow{f}(\overrightarrow{x}_i) - \overrightarrow{f}(\overrightarrow{x}_p)\|}{\sum_{j=1}^{n} \min_{\overrightarrow{x_p}\in\tilde{S}} \|\overrightarrow{f}(\overrightarrow{x}_j) - \overrightarrow{f}(\overrightarrow{x}_p)\|}),\ i = 1, \cdots, n \tag{11}$$

Using (11) the range between poor and better performing particles is extended, as the weaker particles diminish their charges and the stronger ones increase theirs.

In addition, the movement expression (10) has also been changed. In MOEM, the individual movement may be performed in a different direction from that indicated by the force vector for two reasons. The first reason is: each coordinate of the force vector is multiplied by the corresponding coordinate of a range vector, the limits of which are $\overrightarrow{x}_i$ and one of the variable lower/upper bounds. As the range vector has not all coordinates equal, the direction of the movement is changed. An example of this situation is represented in Fig. 1, where the dashed vectors represent the direction of the force vector. This situation always occurs except when the distance from $\overrightarrow{x}_i$ to the respective bounds is the same in all dimensions. The second reason is that the chosen bound depends on whether the force is positive or negative. There are many situations in which the individual force components have different signs. In such situations different range vectors are used, which deviates the individual from the direction indicated by its force vector. In sum, we conclude that in most cases the direction of movement is different from the direction defined by the force. This may bias the particle movement.

To overcome the limitations of the individual updating process in the MOEM algorithm, a modified update position mechanism has been incorporated in EMOEM. In the new position updating expression, the vector of the allowed range of movement is dropped to guarantee that the movement performed by a particle follows the direction of its force vector. Figure 2 represents an example of the update position mechanism performed by the EMOEM algorithm. Then, in the

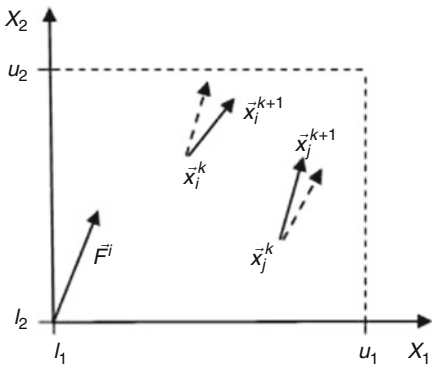


Fig. 1 Update position – MOEM algorithm

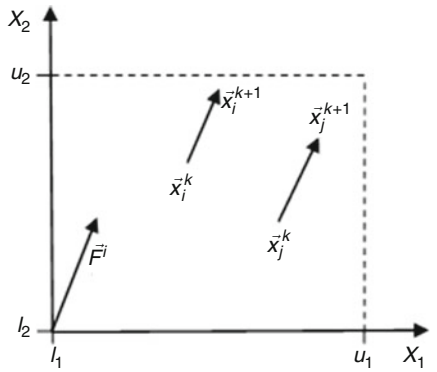


Fig. 2 Update position – EMOEM algorithm

EMOEM algorithm, each particle moves itself according to the expression

$$x_{ir}^{k+1} = x_{ir}^{k} + \lambda \frac{F_r^i}{\|\overrightarrow{F^i}\|},\ r = 1,\ \cdots,\ d \tag{12}$$

where λ is a random uniform value in the interval $[0,\ 1]$. Since the force vector is normalized, variables should be considered in $[0,\ 1]$. To satisfy this requirement, a change of variables is performed. Then, before updating the particle position, its variables are mapped onto the $[0,\ 1]$ interval using the expression

$$x_{ir} \leftarrow \frac{x_{ir} - l_r}{u_r - l_r},\ r = 1,\ \cdots,\ d \tag{13}$$

where l_r and u_r are the lower and upper bounds of variable r, respectively. This ensures that each variable of the particle lies in $[0,\ 1]$ and the particle is ready to be updated. The direction of the movement does not change with the position occupied by the individual in the search space. Then the movement performed in the EMOEM algorithm overcomes the biased situations identified in the MOEM algorithm and at the same time guarantees the feasibility of solutions. In some cases one individual

may become infeasible (i.e., outside the bounds) but in such case it is moved to the corresponding bounds. The implementation and test of this version of EMOEM showed that it performs globally better than MOEM, as it improved the convergence and diversity of the solutions in most problems although not in all. Thus, we propose herein another modification, which consists in an improvement in the local search procedure. Preliminary tests have shown that the new EMOEM version outperforms the previous one. The new local search procedure is described and justified below.

In the electromagnetism-like mechanism, the individual charges and the force vector applied to an individual are affected by all the other individuals. This may prevent the best performing particles from achieving even better positions in the search space. In contrast, local search may provide an exploitation process in which a selected individual is not conditioned by the other individuals. Therefore, the local search procedure is crucial for the success of the algorithm. The MOEM local search strategy presented in Algorithm 4 randomly perturbs each variable of one individual within a predefined step length. If the new solution dominates the current one, this is replaced and the search ends; otherwise, a new random search is performed on the same individual until *lsiter* iterations have been concluded. This situation is illustrated in Fig. 3, where the end point of each vector indicates an alternative solution to be analyzed in terms of dominance with respect to solution $\overrightarrow{x}_i$. Analyzing the behavior of the MOEM algorithm, we found that more often than not the local search produced few new non-dominated solutions. Aiming at improving the MOEM ability to converge to the Pareto front, we introduced a new local search procedure inspired by the Hooke and Jeeves algorithm [9] and its successful application in multiobjective optimization [12]. The proposed local search intends to explore promising directions of search. Taking a selected individual $\overrightarrow{x}_i$ from the local archive, a variable is perturbed each time with a certain probability leading to a new solution. If this trial solution is dominated by $\overrightarrow{x}_i$, or they do not dominate each other, a new perturbation is made. Otherwise, a pattern search is performed from the trial solution following the direction defined by $\overrightarrow{x}_i$ and the trial solution. When this search produces a dominating solution, this is selected to replace $\overrightarrow{x}_i$; otherwise $\overrightarrow{x}_i$ is replaced by the trial solution. An example of this process is illustrated in Fig. 4. Considering the local search starting from solution

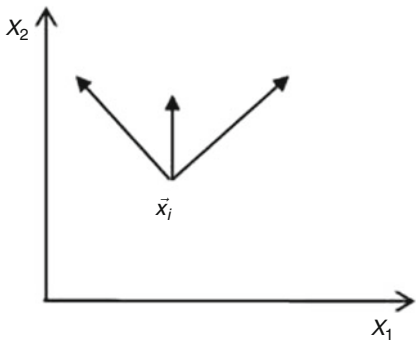


Fig. 3 Local search in MOEM algorithm

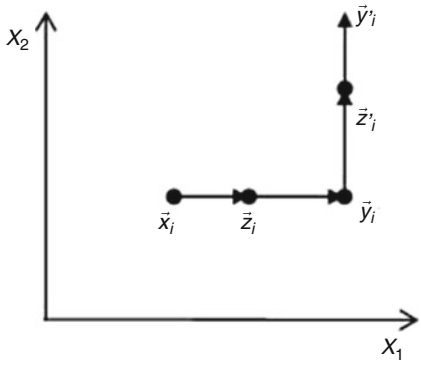


Fig. 4 Local search in EMOEM algorithm

$\overrightarrow{x}_i$, the successful search directions are indicated. By changing variable x_{i1} in $\overrightarrow{x}_i$, $\overrightarrow{z}_i$ is obtained. Considering that $\overrightarrow{z}_i$ dominates $\overrightarrow{x}_i$, then the search proceeds in this direction and $\overrightarrow{y}_i$ is retained considering that it dominates $\overrightarrow{z}_i$. Afterwards, starting from $\overrightarrow{y}_i$, by changing variable y_{i2} in $\overrightarrow{y}_i$, solution $\overrightarrow{z}'_i$ is obtained. The search would still proceed to $\overrightarrow{y}'_i$ if $\overrightarrow{z}'_i$ dominated $\overrightarrow{y}_i$. Considering these two variables of solution $\overrightarrow{x}_i$, four successive non-dominated solutions were found. This illustrates our goal with this strategy, that is to take advantage of a successful direction and pursue the search throughout that direction. The pseudo code of this local search procedure is presented in Algorithm 5.

5 Experimental Results

This section starts by referring to the performance measures used to assess the algorithms. The test problems and the parameter settings of the algorithms are described below. Then, the computational results are shown and analyzed, firstly considering some benchmark multiobjective problems and then an inventory control problem.

5.1 Performance Measures

The multiobjective algorithms are assessed in terms of convergence to the Pareto front and with respect to the diversity of the obtained solutions. In order to assess the proposed EMOEM and compare it with the other algorithms, two unary performance measures commonly used in the literature are considered: Hypervolume indicator (HV), measuring the volume of the space between the non-dominated front obtained and a reference point (usually the nadir point is considered) [8, 23], and the Inverted Generational Distance (IGD), which is the sum of the distances from each

Algorithm 5 Local Search (EMOEM) pseudo code

for $i = 1, \cdots, |\widetilde{S}|$ **do**
  for $k = 1, \cdots, d$ **do**
    $length = \delta \cdot (u_k - l_k)$
    Select a random value $\lambda_1 \in [0, 1]$
    if $\lambda_1 < \frac{1}{3}$ **then**
      Initialize local search counter, $count = 0$
      while $count < lsiter$ **do**
        $\overrightarrow{z} = \overrightarrow{x}_i$
        trial=false
        Select a random value $\lambda_2 \in [0, 1]$
        $z_k = x_{ik} + \lambda_2 \times length$
        if $\overrightarrow{z} \prec \overrightarrow{x}_i$ **then**
          $\overrightarrow{y} = \overrightarrow{z}$
          $y_k = y_k + \delta \cdot (y_k - x_{ik})$
          trial=true
        else
          $z_k = x_{ik} - \lambda_2 \times length$
          if $\overrightarrow{z} \prec \overrightarrow{x}_i$ **then**
            $\overrightarrow{y} = \overrightarrow{z}$
            $y_k = y_k + \delta \cdot (y_k - x_{ik})$
            trial=true
          end if
        end if
        if trial **then**
          if $\overrightarrow{y} \prec \overrightarrow{z}$ **then**
            $\overrightarrow{z} = \overrightarrow{y}$
          end if
          $\overrightarrow{x}_i = \overrightarrow{z}$
          $count = lsiter$
          Update $\widetilde{A}$ with $\overrightarrow{z}$
          $\overrightarrow{x}_i = \overrightarrow{z}$
        else
          $count = count + 1$
        end if
      end while
    end if
  end for
end for

point of the true Pareto front to the nearest point of the non-dominated set found by the algorithm. Both indicators measure the convergence and spread of the set of solutions obtained. The lower the IGD value, the better the approximation is. Larger values of HV indicate better approximation sets.

5.2 Test Problems

Twelve test problems are considered, distributed as follows: 5 bi-objective problems from the test suite ZDT [24]; 3 test problems from DTLZ [6] (dtlz1, dtlz2 and dtlz3 problems), considering three-objective formulations; in the domain of constrained optimization, the 3 bi-objective problems CONSTR, SRN and TNK [5] are considered. The last problem we have selected is an inventory control problem [20] studied in the literature.

5.3 Parameterization of the Algorithms

All the algorithms were implemented in Matlab and the tests were performed on an Intel core 2 Duo 2.4 GHz processor.

Thirty independent runs of each algorithm were performed for each problem. The population size and the non-dominated archive size were set to 100 individuals for all algorithms. In the EMOEM algorithm, the local archive was limited to 5 elements and the number of local search iterations was set to $lsiter = 5$. In order to balance the computational effort of all algorithms, the stop condition for EMOEM algorithm was set to 50000 and 100000 function evaluations, for two and three objective problems, respectively, and for the remaining algorithms, the stop condition was set to 25000 and 50000 function evaluations, for two and three objective problems, respectively. The different values of the stop condition adopted for the EMOEM algorithm are justified by the need of equalizing the computational running cost. It is worthy of mention that approximately half of the function evaluations performed by EMOEM algorithm occur in the local search procedure and the computational cost per function evaluation in this procedure is much lower than in the main cycle of the algorithm.

5.4 Comparing EMOEM, MOEM, OMOPSO, MOSADE and NSGA-II Algorithms

The algorithms used in this comparison are found to be the best performing algorithms of their classes. Table 1 contains the median and inter-quartile range values of the hypervolume obtained by the algorithms in all the benchmark problems. In the problem dtlz1, only the EMOEM and NSGA-II algorithms have obtained solutions inside the hypercube defined by the non-dominated set and the reference point. The same situation occurs in the dtlz3 problem, in spite of the median being zero for the EMOEM algorithm. As can be observed in Table 1, EMOEM ranks second in the performance hierarchy given by the hypervolume for most of the problems. At significance level $\alpha = 0.05$, the differences are statistically significant for all the problems solved. The Kruskall-Wallis test was used.

Table 1 Median and Inter-Quartile Range (IQR) of hypervolume values obtained by EMOEM, MOEM, OMOPSO, MOSADE and NSGA-II algorithms

	EMOEM		MOEM		OMOPSO		MOSADE		NSGA-II	
Problem	Median	IQR	Median	IQR	Median	IQR	Median	IQR	Median	IQR
zdt1	0.6371	0.0149	0.5544	0.1718	0.6312	0.0035	0.6261	0.0019	**0.6571**	0.0007
zdt2	0.3152	0.0129	0.0000	0.0000	0.3010	0.2272	0.2749	0.0050	**0.3248**	0.0005
zdt3	0.7246	0.0618	0.0001	0.1024	0.7020	0.0127	0.7417	0.0031	**0.7736**	0.0006
zdt4	0.5489	0.0242	0.1491	0.0256	0.1010	0.0058	0.1333	0.0163	**0.6569**	0.0013
zdt6	0.0398	0.0185	0.0000	0.0000	0.2681	0.0285	0.0000	0.0000	**0.3162**	0.0007
dtlz1	0.5705	0.5237	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.9683**	0.0018
dtlz2	0.1410	0.0268	0.2176	0.0267	0.3203	0.0106	0.4061	0.0033	**0.3633**	0.0090
dtlz3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.3338**	0.0352
CONSTR	3.5797	0.0550	3.1829	0.0847	2.6038	0.0015	3.5653	0.0500	**5.7584**	0.0032
SRN	**24331**	157.07	23871	297.03	24910	14.799	22096	1561.7	23782	231.02
TNK	0.3187	0.0015	0.3200	0.0029	0.3222	0.0010	0.3239	0.0006	**0.3240**	0.0007

In the case of the IGD performance measure, the results obtained for the five algorithms are reported in Table 2. These results confirm what we have observed with the hypervolume indicator. EMOEM obtains better results than MOSADE in all problems except zdt1, zdt3 and ztlz2, and is better than MOEM in all problems. It can be observed that EMOEM maintains its relative position in constrained problems. The MOEM algorithm presents the worst performance among these algorithms. The results obtained are significant at $\alpha = 0.05$ level for all instances.

The median and inter-quartile range values of execution time spent by each algorithm in the benchmark problems ZDT and DTLZ are reported in Table 3. Although EMOEM algorithm performs more function evaluations, its computational cost still decreases in relation to MOEM algorithm. Generally, the time spent by each algorithm in solving a problem strongly depends on how the algorithm performs in that problem, namely the number of non-dominated solutions obtained, because a better performance of the algorithm generally corresponds to an increase of the size of the non-dominated archive. In many poor performance situations of the algorithms few non-nominated solutions were obtained, and the computational effort decreased substantially. This situation has occurred, for example, in dtlz1 and dtlz3 problems, in MOEM, OMOPSO and MOSADE algorithms.

In general, the NSGA-II algorithm presents better values of the hypervolume and IGD performance measures than the other approaches; however, it also imposes a higher computational effort to achieve a number of non-dominated solutions similar to the one obtained by EMOEM.

5.5 *Inventory Control Problem*

In the economic activity of most companies, the inventory control problem is a critical issue. The problem generally involves the optimization of different conflicting objectives. Most approaches to solve this problem rely on the aggregation of the objectives into a single objective optimization problem. Different multiobjective models for the inventory control problem are described in the literature, depending on the objectives considered to be minimized. Agrell [2] proposed the following multiobjective formulation for the problem:

$$
\begin{aligned}
&Min\,\overrightarrow{f}\,(k,Q) = (C(k,Q), N(k,Q), S(k,Q))\\
&where\\
&C(k,Q) = \tfrac{AD}{Q} + hc(\tfrac{Q}{2} + k\sigma_L)\\
&N(k,Q) = \tfrac{D}{Q}\int_k^{+\infty}\varphi(x)dx\\
&S(k,Q) = \tfrac{D\sigma_L}{Q}\int_k^{+\infty}(x-k)\varphi(x)dx\\
&S.t.\\
&\sqrt{\tfrac{2AD}{hc}} \le Q \le D,\quad 1 \le k \le \tfrac{D}{\sigma_L},
\end{aligned}
\tag{14}
$$

Table 2 Median and Inter-Quartile Range (IQR) of IGD values obtained by EMOEM, MOEM, OMOPSO, MOSADE and NSGA-II algorithms

	EMOEM		MOEM		OMOPSO		MOSADE		NSGA-II	
Problem	Median	IQR	Median	IQR	Median	IQR	Median	IQR	Median	IQR
zdt1	0.0308	0.0256	0.1041	0.1231	0.0215	0.0023	0.0255	0.0014	**0.0064**	0.0003
zdt2	0.0125	0.0086	0.7722	0.1257	0.0206	0.4431	0.0411	0.0051	**0.0063**	0.0004
zdt3	0.0496	0.0394	0.8544	0.6901	0.0377	0.0070	0.0165	0.0013	**0.0066**	0.0005
zdt4	0.0863	0.0288	0.5473	0.0513	0.5564	0.0075	0.5202	0.0203	**0.0065**	0.0004
zdt6	0.5605	0.0686	2.9764	0.6501	0.0236	0.0186	1.2474	0.0751	**0.0065**	0.0008
dtlz1	0.2516	0.5278	1.3785	1.0316	14.884	3.473	10.474	5.1913	**0.0314**	0.0033
dtlz2	0.4645	0.1473	0.1763	0.0205	0.0933	0.0059	0.0714	0.0054	**0.0779**	0.0055
dtlz3	4.5693	12.650	17.482	7.7891	174.98	16.579	180.36	19.424	**0.0886**	0.0253
CONSTR	0.1747	0.0309	1.1218	0.1394	0.8356	0.0004	0.1813	0.0921	**0.1732**	0.0009
SRN	2.2151	0.4660	2.7933	0.7269	**1.0298**	0.0613	8.0903	2.6726	4.4872	0.8205

Table 3 Median and Inter-Quartile Range (IQR) of execution time spent by EMOEM, MOEM, OMOPSO, MOSADE and NSGA-II algorithms

	EMOEM		MOEM		OMOPSO		MOSADE		NSGA-II	
Problem	Median	IQR	Median	IQR	Median	IQR	Median	IQR	Median	IQR
zdt1	160.01	23.954	275.86	4.5162	106.31	3.9780	56.862	2.4414	353.23	2.3127
zdt2	147.66	60.275	297.31	7.5426	111.64	58.048	48.867	4.1223	424.01	1.5561
zdt3	149.84	18.588	271.52	3.4008	59.936	1.7901	45.123	1.5873	440.55	33.080
zdt4	170.66	12.800	203.10	2.0826	31.949	2.5662	17.730	0.0203	420.68	15.296
zdt6	146.033	4.6917	212.47	9.5746	21.965	7.5582	12.347	0.3549	411.30	1.4898
dtlz1	445.30	40.084	204.10	4.0560	14.851	1.0608	42.346	9.6760	816.46	2.5740
dtlz2	699.95	12.862	247.70	1.7667	121.91	2.7378	447.82	183.57	842.17	2.1216
dtlz3	331.34	28.396	214.67	3.7128	20.389	0.7488	42.175	3.3930	813.12	2.3868

Table 4 Inventory Control Problem parameterization

Parameters	A	D	h	c	σ_L
	80	3412	0.26	27.5	53.354

Table 5 Median and Inter-Quartile Range (IQR) of hypervolume values obtained by EMOEM, MOEM, OMOPSO, MOSADE and NSGA-II algorithms

Algorithm	EMOEM	MOEM	OMOPSO	MOSADE	NSGA-II
Median	**86478.8**	82303.9	84771.2	85338.8	85808.8
IQR	**380.62**	4989.15	1350.39	997.56	1308.08

Table 6 Median and Inter-Quartile Range (IQR) of IGD values obtained by EMOEM, MOEM, OMOPSO, MOSADE and NSGA-II algorithms, in the Inventory Control Problem

Algorithm	EMOEM	MOEM	OMOPSO	MOSADE	NSGA-II
Median	**51.6527**	264.301	109.338	84.5601	93.7468
IQR	**11.6312**	267.633	25.9197	18.5143	28.6092

where k and Q are the decision variables that represent the safety factor and the order size; D is the expected annual demand of the product whose unit cost is c, A the fixed ordering cost, h the inventory carrying rate, σ_L the standard deviation of lead time demand, and $\varphi(x)$ the demand density function. The objectives are to minimize the expected total cost, C, the expected number of stockout occasions annually, N, and the expected annual number of the items stocked out, S.

We consider the problem instance addressed in [20], whose parameters are presented in Table 4.

The problem was solved by all the algorithms herein implemented and the hypervolume and IGD measures are reported in Tables 5 and 6, respectively.

Tables 5 and 6 show that EMOEM was the algorithm that provided the best results for this problem.

6 Conclusions and Future Research

Several meta-heuristics have been proposed to address single and multiobjective optimization problems. The electromagnetism-like mechanism (EM) is a relatively recent meta-heuristic that has shown a very good performance in single objective optimization problems [1, 18]. The research work on extending the electromagnetism-like mechanism to multiobjective optimization has been rather scarce. The MOEM algorithm [19] represents a first attempt to use the EM approach in multiobjective optimization. However, the MOEM algorithm has shown

a poor performance in comparison with other state-of-the-art multiobjective metaheuristics.

Motivated by the success of EM in single objective optimization, we have analyzed and modified some main components of the MOEM algorithm leading to an Enhanced MOEM (EMOEM). We have also changed the local search procedure, which further improved the results. The results obtained by the EMOEM algorithm are found to be very competitive when compared with the results of other representative state-of-the-art algorithms (OMOPSO, MOSADE, NSGA-II). In general, EMOEM ranks in the second position, being outperformed only by NSGA-II.

Further research should be pursued in order to make the EMOEM algorithm more robust and competitive, namely for problems with more than two objective functions. Hybridization with other approaches will be studied to enhance EMOEM algorithm performance.

Acknowledgements This R&D work has been partially supported by the Portuguese Foundation for Science and Technology (FCT) under project grant UID/MULTI/00308/2013 and QREN Mais Centro Program Projects EMSURE (CEN- TRO 07 0224 FEDER 002004) and iCIS (CENTRO-07-ST24-FEDER-002003).

References

1. Alikani, M.G., Javadian, N., Tavakkoli-Moghaddan, R.: A novel hybrid approach combining electromagnetism-like method with Solis and Wets local search for continuous optimization problems. J. Glob. Optim. **44**, 227–234 (2009)
2. Agrell, P.J.: A multicriteria framework for inventory control. Int. J. Prod. Econ. **41**, 59–70 (1995)
3. Birbil, S.I., Fang, S.: An electromagnetism-like mechanism for global optimization. J. Glob. Optim. **25**, 263–282 (2003)
4. Carrasqueira, P., Alves, M.J., Antunes, C.H.: An improved multiobjective electromagnetism-like mechanism algorithm. In: Esparcia-Alcázar, A.I., Mora, A.M. (eds.) EvoApplications 2014. LNCS, vol. 8602, pp. 627–638. Springer, Berlin/Heidelberg (2014)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
6. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, L.J.A. (ed.) Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, pp. 105–145. Springer, London (2005)
7. Durillo, J.J., Nieto, J.G., Coello, C.A., Luna, F., Alba, E.: Multi-objective particle swarm optimizers: an experimental comparison. In: 5th International Conference on Evolutionary Multicriterion Optimization (EMO2009), Nantes, pp. 495–509. Springer (2009)
8. Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume. In: Proceedings of 2006 IEEE Congress on Evolutionary Computation, Vancouver, pp. 1157–1163 (2006)
9. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. J. ACM **8**, 212–229 (1961)
10. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Network, Perth, pp. 1942–1948 (1995)

11. Mezura-Montes, E., Reyes-Sierra, M., Coello Coello, C.A.: Multi-objective optimization using differential evolution: a survey of the state-of-the-art. In: Chakraborty, U.K. (ed.) Advances in Differential Evolution, pp. 173–196. Springer, Berlin (2008)
12. Mousa, A.A., El-Shorbagy, M.A., Abd-El-Wahed, W.F.: Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. Swarm Evol. Comput. **3**, 1–14 (2012)
13. Naji-Azimi, Z., Toth, P., Galli, L.: An electromagnetism metaheuristic for the unicost set covering problem. Eur. J. Oper. Res. **205**, 290–300 (2010)
14. Price, K.: Differential evolution vs. the functions of 2nd ICEO. In: IEEE Conference on 15 Evolutionary Computation, Indianapolis, pp. 153–157 (1997)
15. Reyes-Sierra, M., Coello Coello, C.A.: Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance. In: Coello Coello, C.A., Aguirre, A.H., Zitzler, E. (eds.) EMO2005, Guanajuato. LNCS, vol. 3410, pp. 505–519. Springer. (2005)
16. Storn, R., Price, K.: Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report TR-95-012, International Computer Science Institute, Berkeley (1995)
17. Rocha, A.M.A.C., Fernandes, E.M.G.P.: A modified electromagnetism-like algorithm based on a pattern search method. In: Mastorakis, N., Mladenov, V., Kontargyri, V.T. (eds.) Proceedings of the European Computing Conference. Lecture Notes in Electrical Engineering, vol. 2, part 9, chapter 12, pp. 1035–1042. Springer, Berlin/Heidelberg (2009)
18. Tavakkoli-Moghaddam, R., Khalili, M., Naderi, B.: A hybridization of simulated annealing and electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness. Soft Comput. **13**(10), 995–1006 (2009)
19. Tsou, C.-S., Kao, C.-H.: An electromagnetism-like meta-heuristic for multi-objective optimization. In: Proceedings of 2006 IEEE Congress on Evolutionary Computation, Vancouver, pp. 1172–1178 (2006)
20. Tsou, C.-S., Kao, C.-H.: Multi-objective inventory control using electromagnetism-like meta-heuristic. Int. J. Prod. Res. **46**(14), 3859–3874 (2008)
21. Tsou, C.-S., Hsu, C.-H., Yu, F.-J.: Using multi-objective electromagnetism-like optimization to analyze inventory tradeoffs under probabilistic demand. J. Sci. Ind. Res. **67**, 569–573 (2008)
22. Wang, Y.-N., Wu, L.-H., Yuan, X.-F.: Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft Comput. **14**, 193–209 (2010). Springer
23. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
24. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. **8**, 173–195 (2000)
25. Zhang, C., Li, X., Gao, L., Wu, Q.: An improved electromagnetism-like mechanism algorithm for constrained optimization. Expert Syst. Appl. **40**, 5621–5634 (2013)

A Routing/Assignment Problem in Garden Maintenance Services

J. Orestes Cerdeira, Manuel Cruz, and Ana Moura

Abstract We address a routing/assignment problem posed by Neoturf, which is a Portuguese company working in the area of project, building and garden's maintenance. The aim is to define a procedure for scheduling and routing efficiently its clients of garden maintenance services. The company has two teams available throughout the year to handle all the maintenance jobs. Each team consists of two or three employees with a fully-equipped vehicle capable of carrying out every kind of maintenance service. At the beginning of each year, the number and frequency of maintenance interventions to conduct during the year, for each client, are agreed. Time windows are established so that visits to the client should occur only within these periods. There are clients that are supposed to be always served by the same team, but other clients can be served indifferently by any of the two teams. Since clients are geographically spread over a wide region, the total distance traveled while visiting clients is a factor that weighs heavily on the company costs. Neoturf is concerned with reducing these costs, while satisfying agreements with its clients. We give a mixed integer linear programming formulation for the problem, discuss limitations on the size of instances that can be solved to guarantee optimality, present a modification of the Clarke and Wright heuristic for the vehicle routing with time windows, and report preliminary computational results obtained with Neoturf data.

J.O. Cerdeira (✉)
Departamento de Matemática & Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, Portugal
e-mail: jo.cerdeira@fct.unl.pt

M. Cruz
Instituto Superior de Engenharia & Laboratório de Engenharia Matemática, Politécnico do Porto, Porto, Portugal
e-mail: mbc@isep.ipp.pt

A. Moura
Instituto Superior de Engenharia/LEMA, Politécnico do Porto & Centro de Matemática da Universidade do Porto, Porto, Portugal
e-mail: aim@isep.ipp.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_9

1 Introduction

In this paper we address a routing/assignment problem posed by Neoturf, which is a Portuguese company working in the area of project, building and garden's maintenance. One of the services provided by Neoturf is the maintenance of private gardens of residential customers (about 60), whose demands are mainly periodic short time interventions (usually 1–3 hours). In the beginning of each year, the number and the estimated frequency of maintenance interventions to conduct during the year are accorded with each client. That estimate on frequency is then used to settle, in regular conditions, a minimum and maximum periods of time separating two consecutive interventions on the same client. Consecutive days of irregular conditions (e.g., extreme weather conditions) may sporadically change those maximum (or minimum) values.

The amount of work highly depends on seasonality. The company allocates to this service two teams (each consists of two or three employees) during the whole year, which may be reinforced with an additional third team during summer. Each team has a van fully equipped with the tools needed to perform the maintenance jobs. There are customers who should be always served by the same team, while others can be served by any team.

Time windows were established so that visits to the client should occur only within these periods. The clients are geographically spread along an area around Oporto of approximately 10 000 km^2. In 2011, these teams traveled more than 60 000 km, with a significant impact on the costs.

Neoturf aims at finding a procedure to scheduling and routing clients efficiently so to reduce costs, while satisfying the agreements with the clients. The scheduling of clients for each day should be planed on a basis of short periods of time (say ten consecutive working days), since unforeseeable events (e.g., weather conditions, client not available at the time previously arranged) may force to postpone planned interventions and to re-settle the designed scheduling.

The routing of customers in each period is a vehicle routing problem (VRP). VRP designates a large class of problems that deals with the design of optimal routes for fleet of vehicles to serve customers. In part dictated by its practical relevance, VRPs have attracted intense research in Combinatorial Optimization expressed by some thousands of scientific and technical papers covering many aspects of the topic. The books [6, 11, 12] provide an insight into the huge variety of the research on this subject. The basic VRP is the problem of finding a set of routes minimizing the total cost or distance traveled for a number of identical vehicles, located at a depot, to supply a set of geographically dispersed customers with known demands subject to vehicle capacity constraints. A large number of variants and extensions of the basic VRP were proposed to model specific applications, including pickup-and-delivery, stochastic demands, online VRPs, multiple depots, ship routing. The VRP with time windows (VRPTW) is a special case/generalization of VRP where each customer can only be served within established time windows (see [3, 5, 8] for recent surveys on the VRPTW). The problem that we address here is a constrained

version of the VRPTW where (i) some customers, but not all, are to be visited by a certain vehicle (team); (ii) no more than one route is assigned on each day to each vehicle (team) and (iii) each customer that is to be served in each period is assigned to exactly one route, in exactly one day of that period. We give a mixed integer linear programming formulation model for the problem, discuss limitations on the size of instances that can be solved to guarantee optimality, present a modification of the classic Clarke and Wright heuristic for the vehicle routing with time windows [4], and report computational results obtained with Neoturf data.

2 Formulation

We consider the year partitioned into consecutive short periods of time (say 10 consecutive working days) and, for each period P of m consecutive working days, we classify clients as

- mandatory, those for which an intervention has to take place during period P, i.e., the number of days since the last visit till the end of period P exceeds the maximum number of consecutive days which can elapse without any intervention taking place, according to what has been agreed with the client;
- discarded, those for which no intervention is expected to take place during period P, i.e., the number of days since the last visit till the end of period P is lower than the number of consecutive days that were agreed to elapse before a new intervention takes place;
- admissible, those for which an intervention may or may not take place during period P.

Let C be the set of clients to be served in period P. We start with C consisting of all mandatory and admissible clients. If no feasible scheduling is found, the decision maker may consider, among other options, to redefine C removing some or all admissible clients from the period P. If no feasible solution exists even when C only includes mandatory customers, then services to some of these customers have to be postponed to the next period. The customers to be removed from the current period may be selected according to some ranking on customers.

Our problem can be viewed as a VRPTW in which certain clients in C have to be visited by (vehicle) team E_0, other clients have to be visited by team E_1, and the remaining clients can be served indifferently either by team E_0 or by team E_1. We denote the sets of those clients by C_0, C_1 and $C_{0,1}$, respectively.

We based our formulation on the so-called *big M formulation* of the traveling salesman problem with time windows (model 1 in [2]).

We construct a directed weighted graph $G = (V, A, \rho)$ as follows (see Fig. 1). The set of vertices V is equal to $C \cup B$, where each vertex b_i^k of B, with $i = 0, \cdots, m$ and $k = 0, 1$, is the i-th "day (fictitious) copy" of the depot for team E_k. There is an arc (u, v) linking client u to client v if there is any possibility to serve v immediately after visiting u, by a same team. Arcs with both directions link each vertex b_i^k, $i =$

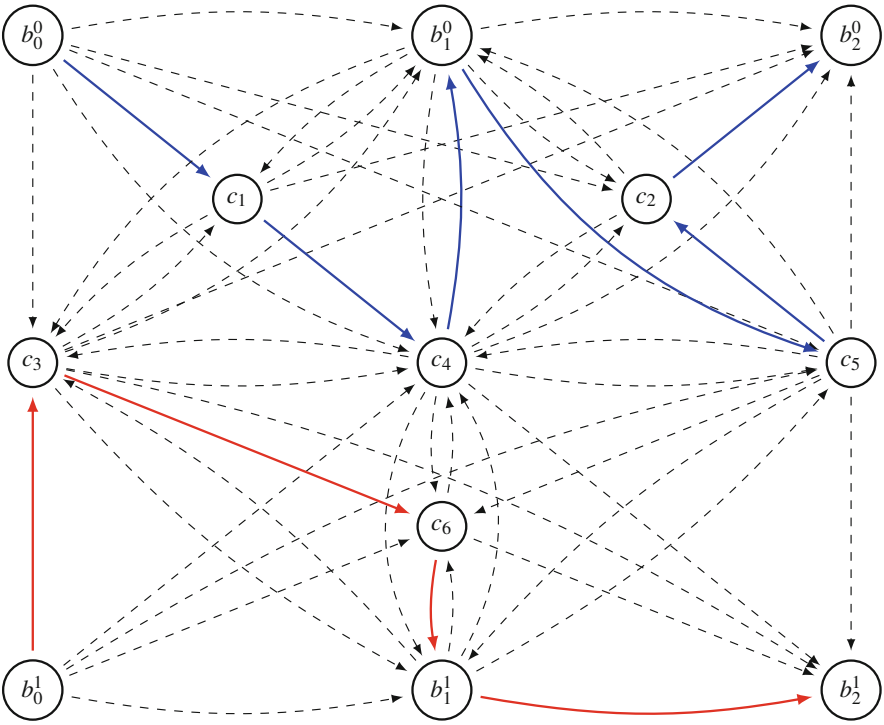


Fig. 1 An example of a directed graph G, and a feasible solution for a two days period. Vertices b_0^0, b_1^0, b_2^0 and b_0^1, b_1^1, b_2^1 are the "fictitious copies" of the depot for team E_0 and team E_1, respectively. The subsets of the set of clients $C = \{c_1, \ldots, c_6\}$ are $C_0 = \{c_1, c_2\}$, $C_1 = \{c_6\}$ and $C_{0,1} = \{c_3, c_4, c_5\}$. The scheduling of clients assigned to team E_0 is represented by the directed path consisting of continuous (*blue*) arcs $Q_0 = (b_0^0, c_1, c_4, b_1^0, c_5, c_2, b_2^0)$. Clients c_1, c_4 and c_5, c_2 are visited by that order on days one and two, respectively. The scheduling of clients assigned to team E_1 is represented by the directed path consisting of continuous (*red*) arcs $Q_1 = (b_0^1, c_3, c_6, b_1^1, b_2^1)$. Clients c_3, c_6 are visited by that order on day one and no client is visited on day two

$1, \cdots, m-1$, with every client of $C_k \cup C_{0,1}$, for $k = 0, 1$. There is an arc from b_0^k to every vertex in $C_k \cup C_{0,1}$, for $k = 0, 1$, but there is no arc with head b_0^k. There is an arc from every vertex in $C_k \cup C_{0,1}$ to b_m^k, $k = 0, 1$, but no arc with tail b_m^k. The other arcs in set A are $(b_0^k, b_1^k), (b_1^k, b_2^k), \cdots, (b_{m-1}^k, b_m^k)$, with $k = 0, 1$, and no more arcs exist linking pairs of vertices in B. For $v \in V$, we use V_v^+ and V_v^- to denote the out-neighborhood and in-neighborhood of v, respectively, i.e., $V_v^+ = \{u \in V : (v, u) \in A\}$ and $V_v^- = \{u \in V : (u, v) \in A\}$.

A scheduling of clients assigned to team E_k will be read on graph G as a directed path Q_k from b_0^k to b_m^k. The clients that are to be visited on day i are the vertices of C on the subpath of Q_k linking b_{i-1}^k to b_i^k. The order of vertices on that path specifies the order by which the corresponding clients should be visited. If arc (b_{i-1}^k, b_i^k) is included in path Q_k it means that no interventions on clients of set C will occur on day i for team E_k.

We define the weight ρ_{uv} of every arc $(u, v) \in A$ as the time to travel on arc (u, v), except when $u, v \in B$, where $\rho_{uv} = 0$.

For each vertex $v \in C$, let $T_v^j = [e_v^j, l_v^j]$ be the j-th time-window of client v, $j = 1, \cdots, nT_v$, where nT_v is the number of time-windows of vertex v, $e_v^j < l_v^j < e_v^{(j+1)}$, and e_v^j and l_v^j are the release time and the deadline time of the j-th time-window of client v, respectively. The release time and deadline time specify minimum and maximum instants for the start of the intervention at the client. For vertices of B, define $T_{b_0^k}^1 = [ST, ST]$ and $T_{b_i^k}^1 = [EN + 24(i-1), EN + 24(i-1)]$, for $i = 1, \cdots, m$ and $k = 0, 1$, where ST and EN are, respectively, the daily service start hour and the daily service end hour.

For $v \in C$, let t_v be the processing time on client v, and set $t_{b_0^k} = 0$ and $t_{b_i^k} = ST + 24 - EN$, for $i = 1, \ldots, m$.

The formulation that we present below uses the following indices, sets, parameters and variables.

Indices

i – days
k – teams
u, v – clients
j – j-th time-windows

Sets

C – clients
C_k – clients to be visited by team k
$C_{0,1}$ – clients served by any of the teams
B – "day (fictitious) copies" of the depot, b_i^k
V – vertices $C \cup B$ of the graph
A – arcs in $V \times V$

Parameters

m – number of days in the period
ρ_{uv} – time to travel on arc (u, v)
t_v - processing time on client v
T_v^j – j-th time-window $[e_v^j, l_v^j]$ of client v
nT_v – number of time windows of client v
e_v^j – release time of the j-th time-window of client v
l_v^j – deadline time of the j-th time-window of client v
ST – daily service start hour
EN – daily service end hour
Δ_{uv} – weight to minimize the number of working days
M - a large number

Variables

x_{uv} – binary variables that are equal to 1 if client v is served immediately after client u, by a same team
y_v^j – binary variables that are equal to 1 if client v is served in time-window T_v^j
a_v – binary variables that assigned client v to team E_{a_v}
s_v – time-instant in which the service starts at client v
w_v – waiting-time to start the service at client v

We deem minimize the sum of travel-time, waiting-time on clients, and number of working days. We thus have the following objective function.

$$\text{Min} \sum_{(u,v)\in A} (\rho_{uv} + \Delta_{uv}) x_{uv} + \sum_{v\in C} w_v \tag{1}$$

where $\Delta_{uv} = EN - ST$ if $u \in C$ and $v = b \in B\backslash\{b_0^0, b_0^1\}$, and $\Delta_{uv} = 0$ for the remaining arcs (u, v), to ensure that optimal solutions will have the minimum number of working days (i.e., the maximum number of arcs (b_{i-1}^k, b_i^k)).

The following equations

$$\sum_{u\in V} x_{vu} = 1, \quad \forall v \in V\backslash\{b_m^0, b_m^1\}, \tag{2}$$

$$\sum_{u\in V} x_{uv} = 1, \quad \forall v \in V\backslash\{b_0^0, b_0^1\}, \tag{3}$$

ensure there will be exactly one arc leaving every vertex $v \neq b_m^k$, and exactly one arc entering every vertex $v \neq b_0^k$.

To force that each client is visited exactly in one of its time-windows, we add equations

$$\sum_{j\leq nT_v} y_v^j = 1, \quad \forall v \in V. \tag{4}$$

To guarantee that the start time occurs within the selected time-window and that vehicle has enough time to travel from u to v, we use the following constraints

$$\sum_{j\leq nT_v} e_v^j y_v^j \leq s_v \leq \sum_{j\leq nT_v} l_v^j y_v^j, \quad \forall v \in V, \tag{5}$$

$$s_u + t_u + \rho_{uv} - (1 - x_{uv})M \leq s_v, \quad \forall (u, v) \in A, \tag{6}$$

where $M > 0$ is large enough (say $M = 24m$) to guarantee that the left hand side is non positive whenever $x_{uv} = 0$, and thus making constraint (6) not active when $x_{uv} = 0$.

Note that constraints (2), (3) together with (6), ensure that the set of selected arcs defines a directed path linking b_0^k to b_m^k, for $k = 0, 1$, where every vertex of V is included exactly once in exactly one of the two paths.

The following inequalities define upper bounds on the waiting-times on clients.

$$w_v \geq s_v - (s_u + t_u + \rho_{uv}) - (1 - x_{uv})M, \quad \forall (u, v) \in A, v \in C, \tag{7}$$

where $M > 0$ is large enough (say $M = 24m$) to guarantee that the right hand side is non positive whenever $x_{uv} = 0$, thus turning the constraint (7) redundant when $x_{uv} = 0$.

The following conditions guarantee that the team assigned to every client v in $C_{0,1}$ is the same team that has visited vertex u, whenever arc (u, v) is in the solution.

$$a_v \leq 1 - x_{uv} + a_u, \quad \forall (u, v) \in A \tag{8}$$

$$a_v \geq x_{uv} - 1 + a_u, \quad \forall (u, v) \in A \tag{9}$$

$$a_v = k, \quad \forall v \in C_k \cup \{b_0^k, b_1^k, \ldots, b_m^k\},\ k = 0, 1 \tag{10}$$

Indeed, if $x_{uv} = 1$, $a_v = a_u$, and if $x_{uv} = 0$, the inequalities (8) and (9) are redundant.

The range of the variables is established as follows.

$$a_v \in \{0, 1\}, \quad \forall v \in C_{0,1} \tag{11}$$

$$x_{uv} \in \{0, 1\}, \quad \forall (u, v) \in A \tag{12}$$

$$y_v^j \in \{0, 1\}, \quad \forall v \in V, \text{ and } j \leq nT_v \tag{13}$$

$$s_v \geq 0, \quad \forall v \in V \tag{14}$$

$$w_v \geq 0, \quad \forall v \in C \tag{15}$$

The above model (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), and (15) gives a mixed integer linear programming formulation for the problem of routing clients of C on a given period of m days, by two teams. The objective function (1) was defined to minimize travel-time and waiting-time on clients in the minimal number of days. Other alternative goals could be considered. For instance, minimizing the total completion-time, i.e., the time of the last service on period P. This could be achieved introducing variable F, imposing the constraints $F \geq s_v + t_v, \forall v \in C$, and defining as objective function: min F. This would give solutions with a minimum number of consecutive working days, and leaving the non working days, if any, at the end of period P. Solutions that define a sequence of consecutive non working days finishing at the end of period P permit to anticipate the next period. However, the objective function (1) expresses the goals specified by Neoturf. The existence of intermittent non working days is not a issue for Neoturf, as it permits to assign the members of the team to other activities.

3 Heuristic Approach

Given the limitations on the size of the instances that could be solved exactly with the formulation (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), and (15) above (see Sect. 4 below), we decided to waive from optimality guaranteed, and use an implementation of Clarke and Wright (C&W) [4] heuristic for the vehicle routing problem with multiple time windows (VRPMTW) available in MATLAB [9].

There are two main issues in applying C&W heuristic to our problem. First, C&W algorithm does not distinguish between clients from C_0, C_1 and $C_{0,1}$. Thus, solutions may include in the same routes clients from C_0 together with clients from C_1.

The second issue follows from the assumption behind C&W algorithm that there are enough vehicles available for the routes determined by the algorithm. Thus, the same team may be assigned, on the same day, to more than one route with incompatible time windows (i.e., services to clients in different routes overlap in time).

To handle the first issue we proceeded as follows.

- We duplicated the number m of days of period P.
- For all clients in C_1, we added $24 \times m$ hours to the release and deadline times of every time-window.
- For all clients in $C_{0,1}$ we duplicated the number of time-windows and, beside the original ones, we also added $24 \times m$ hours to the release and deadline times of every original time window.

Since each client is visited exactly once, in the whole period (now with $2m$ days), within one of its time-windows, setting the time windows of clients C_0 on the first m days and the time-windows of clients C_1 on days $m+1$ to $2m$, ensures that clients from C_0 will not be put together in the same routes with clients from C_1.

Duplicating as described above the number of time-windows of clients $C_{0,1}$, and given that each will be served exactly once, defines a partition of these clients into those that will be served in the first m days (together with clients of C_0), and those that will be served in days $m+1$ to $2m$ (together with clients of $C_{0,1}$).

To address the second issue we use matchings in bipartite graphs.

Suppose the number of routes assigned to a team is less than or equal to m, and there is more than one route on the same day. We consider a bipartite graph (see Fig. 2) where vertices of bi-class R represent routes and vertices of the other bi-class D represent the m days. There is an edge $[r, d]$, with $r \in R$ and $d \in D$, if and only if route r can be done (w.r.t time-windows) in day d.

We then find the maximum matching [7] of this graph. If it has $|R|$ edges, then it indicates how routes should be distributed by the m days of the period, with no more than one route per day. If the maximum matching has less than $|R|$ edges, or $|R| > m$, we propose that the decision maker considers: assigning an extra-team for this period; increasing the number of days in the forecast period, or reducing

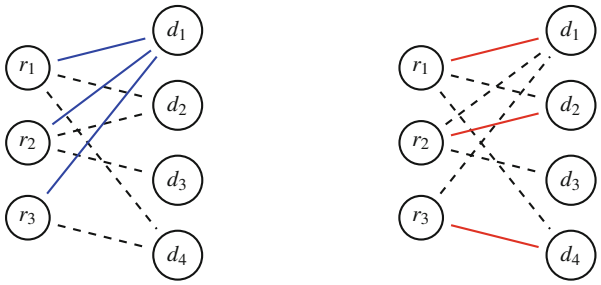


Fig. 2 Bipartite graph with $|R| = 3$ routes assigned to the same team for a period of four days. *Blue edges* (continuous lines on the left picture) indicate the assignment of routes to days on the solution obtained with the modified C&W heuristic. Edges $[r_i, d_j]$ indicate that route r_i can done (w.r.t time-windows) on day d_j. *Red edges* (continuous lines on the right picture) are the edges of a maximum matching. On the left picture, all routes were assigned to the first day. On the right, the maximum matching (*red edges*) defines a feasible assignment of the three routes to three days of the period

the number of admissible clients for the period, and repeat the whole process. Quite often the matching obtained had cardinality $|R|$, which permitted to distribute the $|R|$ routes by the m days. Only in few cases the number of days and/or the set of customers of the period had to be redefined.

4 Computational Results

Here we report some computational experiments carried out with Neoturf data. We call total time to the sum of travel-time and waiting-times, i.e., the values of the objective function not accounting for parameters Δ.

We used the NEOS Server [10] platform to test the model (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), and (15). The implementation was made in AMPL [1] modeling language and ran using the commercial solver Gurobi. On the tests that we carried out, only for periods not exceeding five days Gurobi produced the optimal solutions. On two instances with periods of five days and thirteen customers, with $|C_{0,1}| = 2$ in one instance, and $|C_{0,1}| = 3$ in the other instance, the optimal solutions were obtained. However, on an instance with all parameters with the same size except $|C_{0,1}| = 4$, NEOS Server returned either "timeout" or "out of memory". The same happen for all the instances that we considered with periods of more than five consecutive working days, and no improvements were achieved when we used different parameterization on *threads*, *mipgap* or *timelimit*.

For the small instances for which Gurobi determined optimal solutions, the gap of total routing times of the solutions obtained with C&W heuristic w.r.t. the optimal values (i.e., (T(C&W)-OPT)/OPT, where T(C&W) and OPT are the total time of the

solution obtained with C&W heuristic and the optimal total time, respectively) did not exceed 5 %.

We then compared the planning that Neoturf had established for a 14 days period (18-Feb-2013 till 3-Mar-2013) with the one produced with C&W heuristic. The solution produced with C&W has an total time of 8h54m (waiting-time = 0h00, and 105h24m if working time is also considered) to serve the 27 clients in 7 and 9 working days for teams E_0 and E_1, respectively. The planning of Neoturf consisted of 14h02m total time (waiting-time=1h00, and 110h32m considering working time), 8 days for team E_0 and 11 days for team E_1.

This gives a reduction on total time $(100 \times (14\text{h}02\text{m} - 8\text{h}54\text{m})/14\text{h}02\text{m})$ around 37 %, that significantly decreases costs resulting from distances traveled, specially because the two teams travel around 60 000 km/year.

5 Conclusion

We considered a routing/location problem arising in the context of garden maintenance services. For each day of each period of time (consisting of some consecutive working days) routes are to be designed, starting and ending at a same point, so that every customer is visited only once during that period, by exactly one vehicle and within predefined time-windows. Customers may require a fixed team or be assigned indifferently to any team.

For this new variant of the VRPTW we constructed a directed graph and presented a compact formulation to minimize travel-time and waiting-time on clients that consists of finding vertex-independent paths of the graph, where every vertex is included in exactly one path, and vertices representing customers that require the same team are included in the same path.

The computational tests that we carried out showed that only for periods not exceeding five days we could obtain the optimal solutions. To deal with this limitation we presented a heuristic approach that uses an adaptation of the classic Clarke and Wright (C&W) heuristic for the VRPTW followed by a procedure to find a maximum matching in a bipartite graphs. The adaptation of the C&W heuristic was devised to satisfy the constraint that customers will be served by the team they required. The maximum matching will check, and possibly repair, infeasibilities on the solution obtained from the C&W heuristics regarding the existence of more than one route assigned to the same vehicle, in the same day. The procedure ran quickly on data provide by Neoturf and the solutions produced significantly improved the solutions that were conceived and implemented by Neoturf. Yet we believe that results may be improved using heuristics for routing more sophisticated than C&W, and exploring models alternative to (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), and (15). We intend to pursuit on this direction.

Acknowledgements The problem addressed in this paper was presented by Neoturf at the 86th European Study Group with Industry, held at ISEP/IPP, School of Engineering, Polytechnic of Porto, 7–11 May 2012. The authors are grateful to Neoturf for providing data and for feedback on results.

The authors were supported by the Portuguese Foundation for Science and Technology (FCT). J. O. Cerdeira was funded through the project UID/MAT/00297/2013, CMA (Centro de Matemática Aplicada). M. Cruz was supported by *Laboratório de Engenharia Matemática*. A. Moura was funded through the project UID/MAT/00144/2013 of CMUP (Centro de Matemática da Universidade do Porto).

References

1. AMPL modelling language, Mar 2013. Available online at: http://www.ampl.com/
2. Ascheuer, N., Fischetti, M., Grötschel, M.: Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. Math. Progr. Ser. **A 90**, 475–506 (2001)
3. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, Part I: route construction and local search algorithms. Trans. Sci. **39**(1), 104–118 (2005)
4. Clarke, G., Wright, J.W.: Scheduling of vehicles from a Central Depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
5. Desaulniers, G., Madsen, O.B.G., Ropke, S.: Vehicle routing problems with time windows. In: Toth, P., Vigo, D. (eds.) Vehicle Routing: Problems, Methods, and Applications, 2nd edn. MOS-SIAM Series on Optimization, pp. 119–159. SIAM, Philadelphia (2014)
6. Golden, B.L., Raghavan, S., Wasil, E.A.: The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, New York (2008)
7. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. **2**(4), 225–231 (1973)
8. Kumar, S.N., Panneerselvam, R.: A survey on the vehicle routing problem and its variants. Intell. Inf. Manag. **4**, 66–74 (2012)
9. MATLAB, The MathWorks, Inc., Natick, (2010). http://www.mathworks.com/products/matlab/
10. NEOS Server, Mar 2013. Available online at: http://www.neos-server.org/
11. Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (2002)
12. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications, 2nd edn. MOS-SIAM Series on Optimization. SIAM, Philadelphia (2014)

A Column Generation Approach to the Discrete Lot Sizing and Scheduling Problem on Parallel Machines

António J.S.T. Duarte and J.M.V. Valério de Carvalho

Abstract In this work, we study the discrete lot sizing and scheduling problem (DSLP) in identical parallel resources with (sequence-independent) setup costs and inventory holding costs. We propose a Dantzig-Wolfe decomposition of a known formulation and describe a branch-and-price and column generation procedure to solve the problem to optimality. The results show that the lower bounds provided by the reformulated model are stronger than the lower bounds provided by the linear programming (LP) relaxation of the original model.

1 Introduction

Since the introductory work of Wagner and Whitin [12] a great amount of research has been done on the discrete lot sizing and scheduling problem (DLSP). The original model has been extended from single-item to multiple-item and from single resource to multiple-resource configurations. Also, additional constraints and different cost structures have been studied. Other studies aim at proposing and/or strengthening compact mixed integer linear (MILP) formulations in order to solve larger and more complex instances. Examples of relevant research works on this problem are [4, 5, 9–11]. Most of the published research for problems with parallel resources is devoted to heuristics.

In this work we propose a Dantzig-Wolfe decomposition to a common integer linear (ILP) formulation and a branch-and-price algorithm to solve the problem to optimality. For the single resource problem a similar column generation approach is presented in [2].

A.J.S.T. Duarte (✉)
UNIAG – Applied Management Research Unit and School of Technology and Management, Polytechnic Institute of Bragança, Campus de Santa Apolónia, Apartado 1134, 5301-857 Bragança, Portugal
e-mail: aduarte@ipb.pt

J.M.V.V. de Carvalho
Departamento de Produção e Sistemas, Campus de Gualtar, Universidade do Minho, 4710-057 Braga, Portugal
e-mail: vc@dps.uminho.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_10

For the parallel resource configurations the authors are not aware of similar approaches, although the used decomposition is very close the one used in[7, 8]. However, on those works, the problem of finding the optimal integer solution was not addressed. Also, the problem does have some similarities with the capacitated lot sizing and scheduling problem for which there is also some published research involving column generation, such as [1, 3]. A relatively recent review of methods for this problem can be found in [6].

In Sect. 2 we provide a formal description of the problem. In Sect. 3 we present a compact original ILP formulation. In Sect. 4 we present a minimum cost flow model that can be used to readily compute upper bounds. In Sect. 5 a Dantzig-Wolfe decomposition for the ILP formulation is proposed along with the resulting master problem and subproblem. In Sect. 6 a dynamic programming approach to the resulting subproblem is presented. Three different branching schemes to solve the problem to optimality are presented in Sect. 7. Finally we present some results showing that the lower bounds provided by the reformulated model are stronger than the lower bounds provided by the linear programming relaxation of the original model.

2 Problem Description

There are R identical parallel resources, indexed with $r = 1, \ldots, R$, I items to be processed, indexed with $i = 1, \ldots, I$, and T discrete and equal periods of time, indexed with $t = 1, \ldots, T$. In each time period, any given machine will be producing one *demand unit* of a given item or will be idle.

Without loss of generality, we define the *demand unit* for a given item as the quantity of that item that is possible to process in one machine during one time period. In practice, this can be seen as a minimum lot size for each item. From this point on, demands will be expressed in integer demand units.

Each item has the following associated coefficients: a vector of demands along the planning horizon, $d_i = \{d_{i1}, \ldots, d_{iT}\}$; a startup cost, s_i, which is the cost of starting the production of a different item in a given resource, which is resource and time independent; an inventory holding cost, h_i, defined as the cost of holding one demand unit of item i over one time period (time independent).

The objective is to decide a production schedule (assigning machines to items over the different time periods) that minimizes the sum of startup and holding costs while meeting the required demands (back-orders are not allowed).

3 ILP Formulation

Because the resources are identical, in our formulation, we use the aggregate variables, as defined in [5]. The complete set of variables is:

x_{it} : number of resources producing item i on period t. Variables x_{i0} are defined in order to account for the number of startups in period 1 and should be made equal to a value that reflects the state of the various resources at the start of period 1;

y_{it} : number of resources where production of item i is started on period t and a startup cost is incurred;
z_{it} : number of demand units of item i carried as inventory from period t to period $t+1$. Variables z_{i0} are defined and should be fixed to reflect the inventory level at the start of period 1.

The complete ILP formulation is the following:

$$\min \quad \sum_{i=1}^{I}\sum_{t=1}^{T}(s_i y_{it} + h_i z_{it}) \tag{1}$$

$$\text{s.t.} \quad z_{i(t-1)} + x_{it} = d_{it} + z_{it} \qquad i = \{1,\ldots,I\},\ t = \{1,\ldots,T\} \tag{2}$$

$$y_{it} \geq x_{it} - x_{i(t-1)} \qquad i = \{1,\ldots,I\},\ t = \{1,\ldots,T\} \tag{3}$$

$$\sum_{i=1}^{I} x_{it} \leq R \qquad t = \{1,\ldots,T\} \tag{4}$$

$$x_{it} \geq 0 \text{ and integer} \qquad i = \{1,\ldots,I\},\ t = \{1,\ldots,T\} \tag{5}$$

$$y_{it} \geq 0 \text{ and integer} \qquad i = \{1,\ldots,I\},\ t = \{1,\ldots,T\} \tag{6}$$

$$z_{it} \geq 0 \text{ and integer} \qquad i = \{1,\ldots,I\},\ t = \{1,\ldots,T\} \tag{7}$$

Note that x_{i0} and z_{i0} are actually constants that reflect the initial state of the resources and the initial inventory levels. From this point on, for simplicity and without loss of generality we will assume these constants to be 0.

The objective function (1) sums the startup costs and the holding inventory costs. Constraints (2) express the inventory balance at each period. Constraints (3) ensure that a startup cost is incurred whenever the number of resources used for a given item increases. Finally, constraints (4) limit the number of resources used in each time period, and constraints (5), (6) and (7) specify the type and limits of the variables.

Using a similar formulation and a standard optimization package on a personal computer, the authors of [5] reported that they could not solve instances with $I = 10$, $R = 2$ and $T = 50$ within 30 minutes of computation. It is clear that solving this formulation directly is not practical, even for small instances.

4 Minimum Cost Flow Formulation

When performing branch-and-bound it is important to be able to compute upper bounds. In this section we propose a minimum cost flow formulation for the DLSP. The formulation is incomplete in the sense that inventory costs are accounted but not the startup costs, which means that the optimal solutions of the network flow problem, when they exist, are feasible to the DLSP, but not guaranteed to be optimal. A similar network for single item problems appears on [13].

Consider the following acyclic directed network. There is one supply node, S, whose supply is equal to RT. Consider also a set of T transshipment nodes, one for

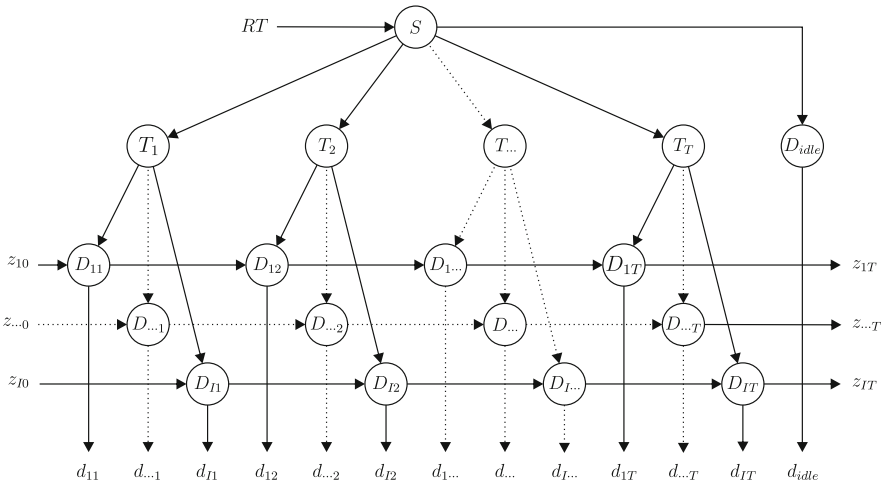


Fig. 1 Minimum cost flow network representation

each time period, named $T_1, \ldots, T_T$. There are arcs from S to T_t with cost 0 and capacity equal to R.

Each of the T_t nodes will be connected to I demand nodes named $D_{1t}, \ldots, D_{It}$. The demand on the D_{it} nodes will be equal to d_{it} and the arcs from T_t to D_{it} have a cost of 0 and unlimited capacity (in practice, the limit will be R). The flow on these arcs has the same meaning as variables x_{it} of the ILP formulation.

Another set of directed arcs will depart from each D_{it} node to the node $D_{i(t+1)}$. These arcs have a cost equal to h_i and unlimited capacity. The flow on these arcs has the same meaning as variables z_{it} of the ILP formulation.

Finally, in order to balance the supply and the demand, consider an additional demand node, D_{idle}, whose demand, d_{idle}, is computed as[1]

$$d_{idle} = RT - \sum_{i=1}^{I} \sum_{t=1}^{T} d_{it}$$

Finally, an arc with cost equal to zero and unlimited capacity, should connect S and D_{idle}. The flow on this arc represents the global capacity excess on the resources.

The complete network is represented on Fig. 1. Note that z_{i0} and z_{iT} can be used to account for, respectively, initial and final inventory levels, if there is need for them to be non-zero.

[1]Note that, if d_{idle} is negative, the problem is infeasible due to a global lack of resource capacity. If d_{idle} is non-negative, the problem can still be infeasible due to demand imbalances over time. A trivial way to check feasibility is to use the same principle to compute the idle capacity at every time period t', i.e., $d^{t'}_{idle} = Rt' - \sum_{i=1}^{I} \sum_{t=1}^{t'} d_{it}$.

Because the flow in arcs (T_t, D_{it}) has the same meaning as variables x_{it} of the ILP formulation, this network can be used to compute feasible solutions to the DLSP that can be used as upper bounds, taking advantage of fast and widely available state-of-the-art minimum cost flow algorithms.

5 Dantzig-Wolfe Decomposition

In this section we apply and present a standard Dantzig-Wolfe decomposition to the ILP formulation presented in Sect. 3.

The ILP formulation has a block angular structure. With the exception of (4), which are coupling constraints, all other constraints can be grouped into I blocks, one for each product item. In our decomposition we will leave constraints (4) in the master problem and group all the constraints that refer to item i to a polyhedron named P_i.

Because any polyhedron P_i is a convex region, any point belonging to P_i can be represented as a convex combination of extreme points. Let p_{ik} be such points. For any P_i polyhedron there will be K_i extreme points, so that $k = 1, \ldots, K_i$. Let $\lambda_{ik} \geq 0$ be the weight of each extreme point in a given combination such that, for any given i, $\sum_{k=1}^{K_i} \lambda_{ik} = 1$. After variable substitution, the master problem will be:

$$\min \quad \sum_{i=1}^{I} \sum_{k=1}^{K_i} c_{ik} \lambda_{ik} \tag{8}$$

$$\text{s.t.} \quad \sum_{i=1}^{I} \sum_{k=1}^{K_i} a_{ikt} \lambda_{ik} \leq R \qquad t = \{1, \ldots, T\} \tag{9}$$

$$\sum_{k=1}^{K_i} \lambda_{ik} = 1 \qquad i = \{1, \ldots, I\} \tag{10}$$

$$\lambda_{ik} \geq 0 \text{ and integer} \qquad i = \{1, \ldots, I\},\ k = \{1, \ldots, K_i\} \tag{11}$$

In this reformulated model, columns can be interpreted as potential schedules for a single item, i, where c_{ik} is the cost of the schedule (including startup and inventory holding costs) and a_{ikt} is number of resources used by the schedule in period t.

Because it is not practical to enumerate all the potential single item schedules, they have to be dynamically generated. Based on the dual solution of the master problem, the subproblems will generate valid and cost attractive schedules to be included in the solution of the master problem.

Each P_i polyhedron will give origin to a different subproblem. Let π_t and ν_i be the dual variables associated with constraints (9) and (10), respectively. Subproblem

i will have the following formulation:

$$\min \quad \sum_{t=1}^{T} (s_i y_{it} + h_i z_{it} - \pi_t x_{it}) - v_i \tag{12}$$

$$\text{s.t.} \quad z_{i(t-1)} + x_{it} = d_{it} + z_{it} \qquad t = \{1, \ldots, T\} \tag{13}$$

$$y_{it} \geq x_{it} - x_{i(t-1)} \qquad t = \{1, \ldots, T\} \tag{14}$$

$$0 \leq x_{it} \leq R \text{ and integer} \qquad t = \{1, \ldots, T\} \tag{15}$$

$$y_{it} \geq 0 \text{ and integer} \qquad t = \{1, \ldots, T\} \tag{16}$$

$$z_{it} \geq 0 \text{ and integer} \qquad t = \{1, \ldots, T\} \tag{17}$$

The subproblem is a single item DLSP on parallel resources. Note that the bounds on x_{it} in constraints (15) are included to avoid the generation of invalid schedules that will never be part of an optimal integer solution to the master problem.

After optimization, for a new column, $c_{ik} = \sum_{t=1}^{T} (s_i y_{it} + h_i z_{it})$ and, hence, the subproblem optimal objective function value is the reduced cost of that column. A generated column is added to the master problem, only if its reduced cost is negative. Also, coefficients a_{ikt} of the new column are equal to x_{it}.

Clearly, if the solution of the reformulated model has only integer variables, then an integer solution to DLSP can be computed. Nevertheless, one relevant characteristic of this problem is that an integer solution to DLSP can also be computed from non-integer variables of the reformulated model, whenever the solution of the reformulated model corresponds to an integer solution in the space of the original variables. This is fully exploited in the branch-and-price algorithm, because the solution in the space of the original variables has to be computed to derive the branching constraints; the branching scheme is presented in Sect. 7.

The following proposition defines the set of conditions that a solution to the master problem must possess in order to be an integer solution to the DLSP:

Proposition 1 *For a solution to the DLSP problem to be integer, it is sufficient that all λ_{ik} variables are integer or that all x_{it} variables are integer, with*

$$x_{it} = \sum_{k=1}^{K_i} a_{ikt} \lambda_{ik} \tag{18}$$

Proof The variables λ_{ik} are binary variables that represent a single item schedule among all the resources, and, if they are all integer, they represent a valid solution. Variables x_{it} are the original formulation variables that represent the number of resources used by item i in time period t. Thus, if all x_{it} are integer, they represent a valid solution.

Consider a new free decision variable, y'_{it} defined as $y'_{it} = x_{it} - x_{i(t-1)}$. This decision variable represents the change in the number of resources producing item i

from period $t-1$ to period t. If there is an increase in the number of resources used, y'_{it} will be positive (equal to the formerly defined y_{it}) and, if there is a decrease, it will be negative. Given this definition, the following proposition is also true:

Proposition 2 *Given the sets of variables x_{it}, y'_{it} and z_{it}, if one of those sets is integer, then, the others must also be integer.*

Proof Variables y'_{it} represent the variation in the number of used resources for a given item and can be computed from x_{it} as stated above. Hence if one of the sets is integer the other is also integer. Variables z_{it} are inventory levels and so $z_{it} = z_{i(t-1)} + x_{it} - d_{it}$. Because d_{it} are integer values, the previous reasoning still applies.

6 Subproblem Optimization

In this section we present a dynamic programming algorithm to solve the subproblem, a single item DLSP. The algorithm evaluates function $F_t(z, r)$ that represents the minimum cost to get z inventory level at the end of period t with r resources setup for the production of the considered item. If we assume that all resources are idle at instant 0, and the initial inventory is 0, then, $F_0(0, 0) = 0$. At each stage transition, we must decide how many resources will be allocated to the production of the considered item, i. Let $x_{it} \in \{0, \ldots, R\}$ be that value. Then, from state (z, r) at stage $t-1$ we can reach, at stage t, states $(z'z - d_{it} + x_{it}, r' = x_{it})$ as long as $z' \geq 0$, because inventory can not be negative. The objective function will be computed in the following way:

$$F_t(z', r') = \begin{cases} F_{t-1}(z, r) - \pi_t r' + h_i z' + s_i(r' - r) & \text{if } r' > r \\ F_{t-1}(z, r) - \pi_t r' + h_i z' & \text{if } r' \leq r \end{cases} \tag{19}$$

At each stage, the maximum theoretical number of states will be equal to $(R + 1)(z_t^+ - z_t^- + 1)$, where z_t^- and z_t^+ are bounds on the inventory level at the end of period t and can be computed as follows:

$$z_t^- = \max\left(0, d_{i(t+1)} - R + z_{t+1}^-\right) \tag{20}$$

$$z_t^+ = \min\left(\sum_{l=1}^{t}(R - d_{il}), \sum_{l=t+1}^{T} d_{il}\right) \tag{21}$$

In Eq. (20) computation is recursive and should be initialized with $z_T^- = 0$, stating that the minimum inventory at the end of period T should be 0 (see discussion on Sect. 4). The computations reflect the fact that, when the demand exceeds R, there will be need for inventory at the end of the previous period or periods.

Concerning the Eq. (21), the maximum inventory is the minimum value between the achievable inventory at the end of period t using maximum capacity and the maximum inventory needs to satisfy demand from inventory for the rest of the planning horizon (once again, assuming that the final inventory should be 0).

Note that these bounds can be used to improve (7) in the ILP formulation and (17) in the subproblem formulation and can be easily modified in the presence of initial and final inventories.

The above mentioned number of states is the theoretical maximum because if, for some state, $F_t(z, r)$ equals or exceeds v_i, further transitions from that state can be ignored, because the reduced cost of the new column would not be negative and, hence, the column would not be attractive.

7 Branching

Solving the relaxed master problem to optimality does not guarantee an integer solution. For that reason, in order to find an integer optimal solution it is necessary to identify and eliminate fractional solutions. Branching is a standard procedure to achieve that goal.

As it is widely known, when performing column generation, branching on the master problem variables (λ_{ik}) is not a good idea, because it leads to column regeneration whenever a branching decision of the type $\lambda_{ik} \leq 0$ is made.

Given Proposition 2, presented in Sect. 5, the sets x_{it}, y'_{it} and z_{it} are natural candidates for branching. The choice should be made based on the results of computational performance tests.

Note that the original variables y_{it} cannot be used for branching because, although integrality on x_{it} implies integrality on y_{it}, the converse is not true. For example, consider the number of resources used (a_{ikt} vectors) in two four-period schedules for a given item: (0,4,4,4) and (4,4,3,1). Suppose that, in the optimal solution of a given node, both λ_{ik} are at a level of 0.5. As it can be easily seen, $x_{ik} = (2, 4, 3.5, 2.5)$ while $y_{ik} = (2, 2, 0, 0)$. This solution would be fractional, while the y_{it} vector would be integer. In this case, the vector y'_{it} would be $(2, 2, -0.5, -1)$ and, hence, not integer.

The following subsections present the 3 possible branching schemes along with the adjustments to the subproblem structure.

7.1 Branching on x_{it}

When branching upon the x_{it} variables, in node j, two branches of the problem are created. On one branch (the left branch) the constraint

$$x_{it} \leq \lfloor x_j^* \rfloor \tag{22}$$

is added, where x_j^* represents some non-integer value. On the other branch (the right branch) the following constraint is added instead:

$$x_{it} \geq \lceil x_j^* \rceil \tag{23}$$

With respect to finding the optimal solution of the model at a given node j, it is necessary to call the subproblems for attractive columns not yet included in the master problem. In node j, besides the initial constraints, the master problem has other sets of constraints, denoted as P^j_{it}, with $i = 1, \ldots, I$ and $t = 1, \ldots, T$, resulting from all the branching decisions imposed on each different variable x_{it}.

Let $\rho^p_{it,j}$ be the dual variable associated with constraint p, with $p \in P^j_{it}$. Thus, in order for the subproblem to correctly identify the attractive columns, in the objective function (12) and in the recursive equation (19), π_t must be replaced with $(\pi_t + \rho^j_{it})$, where ρ^j_{it} is the sum of all dual variables, $\rho^p_{it,j}$, associated with constraints $p \in P^j_{it}$, which are imposed on the variable x_{it} at node j, i.e., $\rho^j_{it} = \sum_{p \in P^j_{it}} \rho^p_{it,j}$.

7.2 Branching on z_{it}

Branching on the z_{it} variables requires some additional manipulations. Developing $z_{it} = z_{i(t-1)} + x_{it} - d_{it}$ recursively yields the following (assuming the starting inventory is 0):

$$z_{it} = \sum_{l=1}^{t} (x_{il} - d_{il}) \tag{24}$$

To translate z_{it} to the master problem space, once again, Eq. (18) should be used. Using the same approach as before, on node j we want to branch on variable z_{it}, whose fractional value is z^*_j. The left and right branching constraints will be, respectively:

$$z_{it} \leq \lfloor z^*_j \rfloor \tag{25}$$

$$z_{it} \geq \lceil z^*_j \rceil \tag{26}$$

Using the same notation as in Sect. 7.1, if ρ^j_{it} is the sum of the dual variables that refer to constraints imposed on the variable z_{it}, the modification to objective function (12) and to the recursive equation (19) is the replacement of h_i by $(h_i - \rho^j_{it})$.

7.3 Branching on y'_{it}

Let y'^*_j be the fractional value of y'_{it} that we wish to branch upon on node j. The constraints to impose on the left and right branches are, respectively,

$$y'_{it} \leq \lfloor y'^*_j \rfloor \tag{27}$$

$$y'_{it} \geq \lceil y'^*_j \rceil \tag{28}$$

On these equations, y'_{it} can be replaced with $x_{it} - x_{i(t-1)}$ and projected to the master problem space using Eq. (18). Once again, as in the previous sections, let ρ^j_{it} be the sum of the dual variables whose associated constraints refer to variable y'_{it}.

In this case, the modifications to the subproblem structure are more complex than in the previous branching schemes presented on Sects. 7.1 and 7.2.

In the case of the ILP formulation there is the need of creating a set of variables to account for decreases in the number of used resources. Let's name those variables y^-_{it}. In the objective function (12) a new term associated with this new variables must be included rendering the following objective function:

$$\sum_{t=1}^{T} \left((s_i - \rho^j_{it}) y_{it} + h_i z_{it} - \pi_t x_{it} + \rho^j_{it} y^-_{it} \right) - v_i \tag{29}$$

Also, an additional set of constraints must be included (similar to constraints (14)):

$$y^-_{it} \geq x_{i(t-1)} - x_{it} \qquad t = \{1, \ldots, T\} \tag{30}$$

Also, in the subproblem formulation that resulted from the decomposition, the y_{it} variables have no upper bound because it is implicitly assumed that their coefficients on the objective function are always positive. Because this last assumption is no longer true, an upper bound on y_{it} equal to $\max\left(0,\ x_{it} - x_{i(t-1)}\right)$ must be enforced in the ILP subproblem formulation. The same logic applies to the y^-_{it} variables: an upper bound equal to $\max\left(0,\ x_{i(t-1)} - x_{it}\right)$ must be enforced. For simplicity, the necessary additional constraints are omitted here.

The recursive equation (19) needs also to be modified and, after the necessary modifications, it will be:

$$F_t(z', r') = \begin{cases} F_{t-1}(z, r) - \pi_t r' + h_i z' + (s_i - \rho^j_{it})(r' - r) & \text{if } r' > r \\ F_{t-1}(z, r) - \pi_t r' + h_i z' + \rho^j_{it}(r - r') & \text{if } r' \leq r \end{cases} \tag{31}$$

With this changes, the subproblem will correctly process the additional dual information.

8 Computational Results

In order to access the quality of our approach, an implementation was developed in C# (Microsoft .NET framework 4.5) using ILOG CPLEX 12.5.0.1 for optimization, with the default parameters. All tests were run in a laptop with a Intel Core i7 3610QM @ 2.30 GHz CPU. The branching scheme is based on the x_{it} variables, as described in Sect. 7.1. This choice was made based on the performance results

of a limited set of preliminary computational tests, which pointed towards a better performance of the partition scheme based on the x_{it} variables.

The test instances were generated randomly, using the procedure described in [5]. Namely, the inventory holding costs (h_i) come from an integer Uniform distribution between 5 and 10, the startup costs (s_i) come from an integer Uniform distribution between 100 and 200 and the demands for a randomly chosen set of (i, t) pairs (d_{it}), come from an integer Uniform distribution between 1 and R. Furthermore, the instances have similar characteristics, namely, there are 4 sets of instances:

- set A: small instances ($R = 2$, $I = 10$ and $T = 50$);
- set B: instances with a large number of periods ($R = 2$, $I = 10$ and $T = 150$);
- set C: instances with a large number of items ($R = 2$, $I = 25$ and $T = 50$);
- set D: instances with a large number of resources ($R = 10$, $I = 10$ and $T = 50$).

These sets were combined with 5 levels of used capacity (75 %, 80 %, 85 %, 90 % and 95 %). For each combination, 3 instances were generated, resulting in a total of 60 instances.

The computational results are shown in Table 1, where each line contains aggregate results for the 3 instances in each combination described above, and the columns have the following meaning: column *UC* refers to the used capacity; columns *Nodes* and *Cols* are the average number of nodes in the branch-and-price tree and the average number of columns generated, respectively; columns *TMIP* and *TBP* are average times (in seconds) to solve to optimality the ILP formulation presented in Sect. 3 (*TMIP*) using the CPLEX MIP Solver and the proposed branch-and-price framework (*TBP*), respectively; columns *SMIP* and *SBP* show the number of instances solved to optimality using each procedure within a time limit of 30 minutes; column *LBInc* shows the average increase, in percentage of the ILP formulation LP relaxation bound, to the LP relaxation of the reformulated model[2]; finally, column *Gap* shows the average gap, in percentage, between the LP relaxation of the root node and the optimal (or best) integer solution found.[3]

In addition to this set of results, we also tested our approach with the instances used in [5]. These results appear in Table 2. The instances are similar to the generated ones with the exception that, instead of 3 instances per combination of parameters, there are 5 instances per combination.[4]

The most noticeable result in the presented tables is that, for every set of instances, except for set D, the computational times are faster than the ones obtained with the CPLEX MIP solver. As noticeable, only for the instances in set D, has our approach a poorer performance, which seems to indicate that it is not so well suited

[2] Let *ILPRel* be the optimal objective value for the ILP relaxation and *RMRel* be the optimal objective value for the linear relaxation of the reformulated model (relaxation of the search tree root node). Using the above notation, $LBInc = 100 \times (RMRel - ILPRel) / ILPRel$.

[3] If *Best* represents the optimal or best integer solution found, $Gap = 100 \times (Best - RMRel) / RMRel$.

[4] Except for set C (instances with 75 % used capacity) where only 4 instances were available.

Table 1 Computational results

Instance set	UC	Nodes	Cols	TMIP	SMIP	TBP	SBP	LBInc	Gap
A: $R = 2, I = 10$ and $T = 50$	75	1.7	602.3	2.36	3	0.39	3	89.4	0.01
	80	3.0	512.0	0.90	3	0.55	3	70.0	0.06
	85	1.0	774.0	4.09	3	0.45	3	85.6	0.00
	90	5.7	1031.0	1.20	3	0.41	3	67.0	0.16
	95	34.0	1686.0	4.85	3	0.66	3	69.0	0.32
B: $R = 2, I = 10$ and $T = 150$	75	251.3	4795.7	219.47	2	14.42	3	87.3	0.30
	80	3844.3	22,495.0	–	0	255.79	3	96.4	0.26
	85	2051.3	35,514.0	–	0	19.07	2	75.9	0.43
	90	617.0	18,415.7	–	0	91.90	3	78.8	0.32
	95	2624.0	78,555.7	–	0	1046.95	2	75.6	0.48
C: $R = 2, I = 25$ and $T = 50$	75	3.7	583.0	1.20	3	0.52	3	88.0	0.03
	80	1.0	716.0	3.20	3	0.57	3	90.9	0.00
	85	35.7	1040.7	5.44	3	0.53	3	105.3	0.05
	90	55.7	1010.7	5.09	3	0.50	3	85.9	0.13
	95	700.3	1710.7	4.73	3	1.10	3	71.5	0.18
D: $R = 10, I = 10$ and $T = 50$	75	30.3	573.0	0.99	3	5.60	3	8.9	0.29
	80	5.7	858.3	1.44	3	5.90	3	10.2	0.03
	85	1228.3	3080.0	1.97	3	50.21	3	6.3	0.37
	90	465.3	5410.0	1.94	3	144.79	3	7.8	0.28
	95	3185.0	12,928.3	6.85	3	214.27	2	9.3	0.67

for problems with a high number of resources to be scheduled. On the other hand, for the instances in set B, our approach solved to optimality 32 of the 40 instances, while the MIP solver only solved 2 instances to optimality.

Another important result is the linear relaxation improvement that our decomposition achieves. This improvement is consistent across all instances tested and clearly shows the merits of this approach. The instances in set D are the ones with the lowest increase in the linear relaxation bound, which seems to indicate that, when the number of resources increases, the decomposition is not as effective. This is consistent with the previous paragraph comment.

Another interesting point to notice is the small *Gap* values in the last column of the tables. It means that the linear programming relaxation at the root node provides a very tight lower bound on the optimal integer solution. Even for the cases when our approach fails to solve all the instances to optimality, there is a small gap between the lower bound and the best known solution (e.g. set B with 95 % used capacity).

The computational results in Tables 1 and 2 should be similar and, in fact, they show congruency, although the instances in Table 2 seem to be slightly harder to solve. This could be due to some difference in our interpretation or our implementation of the random generation procedure detailed in [5].

Table 2 Computational results for instances in the literature 2

Instance set	UC	Nodes	Cols	TMIP	SMIP	TBP	SBP	LBInc	Gap
A: $R = 2, I = 10$ and $T = 50$	75	5.2	479.0	2.14	5	0.66	5	95.8	0.26
	80	8.4	572.4	2.35	5	0.72	5	82.0	0.18
	85	29.6	795.6	5.13	5	0.72	5	87.3	0.41
	90	17.8	1124.0	3.25	5	0.80	5	73.2	0.26
	95	30.0	1667.4	16.44	5	0.96	5	70.2	0.45
B: $R = 2, I = 10$ and $T = 150$	75	124.6	3518.2	–	0	25.33	5	116.0	0.18
	80	664.8	10,665.8	–	0	88.03	5	111.9	0.15
	85	1047.4	15,794.2	–	0	116.42	5	102.3	0.44
	90	2704.4	56,567.4	–	0	486.86	4	103.7	0.73
	95	2656.2	86,827.2	–	0	–	0	88.2	1.23
C: $R = 2, I = 25$ and $T = 50$	75	18.0	695.0	5.07	4	0.85	4	122.5	0.09
	80	19.4	772.4	6.28	5	0.90	5	126.3	0.12
	85	89.2	943.6	7.94	5	0.96	5	123.2	0.13
	90	82.0	1213.0	33.58	5	1.09	5	135.2	0.21
	95	320.8	1598.6	19.18	5	1.35	5	112.6	0.54
D: $R = 10, I = 10$ and $T = 50$	75	14.0	421.4	0.65	5	3.14	5	11.4	0.12
	80	30.0	847.2	1.16	5	5.50	5	8.8	0.13
	85	54.8	1208.4	1.08	5	6.72	5	9.7	0.19
	90	2105.2	4618.8	5.53	5	170.97	4	10.2	0.29
	95	5815.0	22,939.4	7.85	5	442.06	1	11.3	0.68

9 Conclusions and Future Work

In this work we presented a column generation approach to a known problem. The computational results show that the presented algorithm can be used with success to solve many real word size instances in very short times. They also show that, when optimality is not achieved, the objective value of the best solution is close to the lower bound provided by our column generation approach.

On the other hand, for some types of instances, with a high number of resources to be scheduled, the results are not so good. Future research efforts should try to fully understand those results and to improve the performance for that set of instances, probably with the help of additional cuts, different branching schemes and/or with an heuristic approach.

Acknowledgements The authors want to thank the anonymous reviewers of the IO2013 conference for the insightful comments to the first version of this paper and the authors of [5] for kindly providing the problem instances they used in their work. This work has been partially supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: PEst-OE/EEI/UI0319/2014.

References

1. Caserta, M., Voß, S.: A math-heuristic Dantzig-Wolfe algorithm for capacitated lot sizing. Ann. Math. Artif. Intell. **69**(2), 207–224 (2013)
2. Cattrysse, D., Salomon, M., Kuik, R., van Wassenhove, L.N.: A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. Manag. Sci. **39**(4), 477–486 (1993)
3. Degraeve, Z., Jans, R.: A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. Oper. Res. **55**(5), 909–920 (2007)
4. Gicquel, C., Minoux, M., Dallery, Y.: Exact solution approaches for the discrete lot-sizing and scheduling problem with parallel resources. Int. J. Prod. Res. **49**(9), 2587–2603 (2011)
5. Gicquel, C., Wolsey, L.A., Minoux, M.: On discrete lot-sizing and scheduling on identical parallel machines. Optim. Lett. **6**(3), 545–557 (2012)
6. Karimi, B., Fatemi Ghomi, S.M.T., Wilson, J.M.: The capacitated lot sizing problem: a review of models and algorithms. Omega **31**(5), 365–378 (2003)
7. Lasdon, L.S., Terjung, R.C.: An efficient algorithm for multi-item scheduling. Oper. Res. **19**(4), 946–969 (1971)
8. Manne, A.S.: Programming of economic lot sizes. Manag. Sci. 4(2):115–135 (1958)
9. van Eijl, C.A., van Hoesel, C.P.M.: On the discrete lot-sizing and scheduling problem with Wagner-Whitin costs. Oper. Res. Lett. **20**(1), 7–13 (1997)
10. van Hoesel, S., Kolen, A.: A linear description of the discrete lot-sizing and scheduling problem. Eur. J. Oper. Res. **75**(2), 342–353 (1994)
11. van Hoesel, S., Wagelmans, A., Kolen, A.: A dual algorithm for the economic lot-sizing problem. Eur. J. Oper. Res. **52**(3), 315–325 (1991)
12. Wagner, H.M., Whitin, T.M.: Dynamic version of the economic lot size model. Manag. Sci. **5**(1), 89–96 (1958)
13. Zangwill, W.I.: Minimum concave cost flows in certain networks. Manag. Sci. **14**(7), 429–450 (1968)

A Tool to Manage Tasks of R&D Projects

Joana Fialho, Pedro Godinho, and João Paulo Costa

Abstract We propose a tool for managing tasks of Research and Development (R&D) projects. We define an R&D project as a network of tasks and we assume that different amounts of resources may be allocated to a task, leading to different costs and different average execution times. The advancement of a task is stochastic, and the management may reallocate resources while the task is being performed, according to its progress. We consider that a strategy for completing a task is a set of rules that define the level of resources to be allocated to the task at each moment. We discuss the evaluation of strategies for completing a task, and we address the problem of finding the optimal strategy. The model herein presented uses real options theory, taking into account operational flexibility, uncertain factors and the task progression. The evaluation procedure should maximize the financial value for the task and give the correspondent strategy to execute it. The procedure and model developed are general enough to apply to a generic task of an R&D project. It is simple and the input parameters can be inferred through company and/or project information.

J. Fialho (✉)
Instituto de Engenharia de Sistemas e Computadores (INESC) Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

CI&DETS and Escola Superior de Tecnologia e Gestão de Viseu, Campus Politécnico, 3504-510 Viseu, Portugal
e-mail: jfialho@estgv.ipv.pt

P. Godinho
Grupo de Estudos Monetários e Financeiros (GEMF), Faculdade de Economia da Universidade de Coimbra, Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal
e-mail: pgodinho@fe.uc.pt

J.P. Costa
Instituto de Engenharia de Sistemas e Computadores (INESC) Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

Faculdade de Economia da Universidade de Coimbra, Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal
e-mail: jpaulo@fe.uc.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_11

1 Introduction

Companies operating in dynamic markets, driven by technological innovation, need to decide, at each moment, which projects to carry out and the amount of resources to allocate to them. These decisions are crucial for the companies' success. Projects that create or improve an existing process, material, device, product or service, or projects that aim to extend overall know-how or ability in a field of science or technology, are considered as research and development (R&D) projects [19]. The nature of this kind of projects may lead to a high cost [19]. Furthermore the R&D outcomes may take years to be realized. Hence, R&D projects should be properly valued and managed, especially for firms that depend on innovation [12]. An effective evaluation of these projects allows allocating the limited resources properly, as well as prioritizing the current projects, according to their expected financial returns.

Traditional evaluation methods, such as the ones based on discounted cash flows, are not adequate because they assume a pre-determined and fixed plan, which does not allow taking into account both uncertainty and flexibility [20]. This kind of methods estimate the future benefits, in terms of cash flows, usually on an annual basis, and then the cash flows are discounted at a risk-adjusted rate so that the present value is obtained. If the initial investment is subtracted, the net present value is obtained [12]. However, predicting future cash flows is not easy in an R&D environment, because the profitability also depends on how the projects are managed during their lifetime [12]. The methods based on discounted cash flows reflect the passive management of a project.

R&D projects are characterized by several types of uncertainty and by the possibility of changing the initial plan of action, that is, R&D projects have two important features that have to be taken into account: uncertainty and operational flexibility. The flexibility of a project leads to an increase in the project value that must be taken into account in its analysis or evaluation. When there is operational flexibility, it may be better to change the plan of action when new information arrives. Hence, it is very important to consider these features in the evaluation of R&D projects.

Mostly, R&D projects are composed by different phases. We can also consider that each project or phase is split into different tasks. Usually, the companies undertaking those projects have different kinds of resources that can be allocated to those tasks. The difference among resources can be qualitative, quantitative, or both. Consequently the cost and the execution speed are different among different levels of resources. Thus, both evaluation and resource allocation have to consider the project flexibility. The flexibility during the execution of a project is very important for seizing opportunities or avoiding losses upon the occurrence of unfavorable or unexpected scenarios [1]. The flexibility can consider different actions at different phases of an R&D project like defer, expand or abandon; but this flexibility is also important to do an active and better resource allocation, that is, the allocation can be changed according to the project progress or the occurrence of unexpected events.

Hence, the flexibility is relevant in order to make optimal decisions, because it leads to an increase in the project value that must be taken into account in its evaluation or management. When new information arrives, the flexibility allows changing the plan of action, if necessary.

In this paper, we intend to present a tool to evaluate tasks of R&D projects, taking into account the resource allocation strategy. For each task, there are different levels of resources that can be chosen. We define a strategy as a set of rules that indicate which level of resources shall be chosen, at each moment, among the levels of resources defined to execute the task.

The main condition to use the model behind the tool is the ability to define a finite, discrete set of levels of resources that can be used at each instant, and to define the cost *per* time unit of each level of resources.

The output of this tool helps management to allocate resources to tasks that compose an R&D project. Although the tool presented evaluates a task of an R&D project, it implies that the evaluation of the tasks that compose a project leads to a financial evaluation of the project. The connection between tasks will be detailed in future work, because it is necessary to determine how the tasks are linked to each other and how codependent they are. Notice that some tasks can have precedents, that is, some tasks can only begin when others are completed or if others have obtained certain results.

For each task, we assume that different resource levels can be allocated, which have different costs and different average execution times. The advance of the task is stochastic and the project manager can reallocate resources while the task is in progress. The progression of a task defines, at each moment, which is the best level of resources to choose. The difference between the resource levels can be quantitative, qualitative or both. Different strategies are analyzed, and the objective is to find the optimal strategy to execute an R&D task.

This paper is structured as follows: Sect. 2 reviews some literature about evaluation and management of R&D projects; Sect. 3 presents and characterizes the model, Sect. 4 describes the analysis procedure, Sect. 5 presents some examples, Sect. 6 describes the usage of the procedure, and Sect. 7 concludes.

2 Evaluation of R&D Projects

There are several models and techniques to evaluate R&D projects and tasks, but it is difficult to aggregate all issues that characterize this kind of projects in a single model.

If the project evaluation is required to be mostly financial, real options theory seems to be quite promising, since it integrates the operational flexibility and the uncertainty into the evaluation process, assisting in the best decisions [19]. A real option gives the right (but not the obligation) to perform a determined action. For example, an R&D laboratory gives to the company the right to research and develop new products but not the obligation to do so [19]. Real options valuation is based on

financial options theory and allows assessing investments under uncertainty, because it takes into account the risks and the flexibility value for making decisions when new information arrives [1].

For the valuation of R&D projects, real options consider that managers have the right but not the obligation to act upon the development process. The evaluation models based on real options emphasize the flexibility and the options available to management [15]. The recognition that the financial options theory can be used to evaluate investment projects was made by Myers [11], who used the expression real option to express the management flexibility under uncertain environments. Real options theory allows us to determine the best sequence of decisions to make in an uncertain environment, and provides the proper way to evaluate a project when such flexibility is present. The decisions are made according to the opportunities that appear along the project lifetime, which means that the optimal decision-path is chosen step by step, switching paths as events and opportunities appear [7].

The models to evaluate real options can present some difficulties like finding the right model, determining the model inputs and being able to solve the option pricing equations [15]. Although some evaluation aspects could be defined through qualitative assessment, it is quite hard to capture interactions among factors or multi period effects during the project. Still, following a real options perspective on R&D projects has a positive impact on both R&D and financial performance.

Many authors use real options to evaluate R&D projects in different areas. Brach and Paxson [2], for example, use real options to evaluate pharmaceutical R&D, Lint and Pennings [9] also use a real options model in the Philips Electronics, Schwartz and Zozaya-Gorostiza [17] use real options in information technology, and Lee and Paxson [6] in e-commerce. However, these models and other similar ones, cannot be flexible enough to adapt to all companies.

To evaluate or analyze an R&D project through real options theory, management has to evaluate the sequential real options that appear along the lifetime of the project. To evaluate these options, it is important to incorporate the associated risk. This risk may be related to prices, costs and technology, among others. There are several processes to model these variables, like Brownian motions [6], mean reversing models [5], controlled diffusion processes [17], or even combinations between diffusion and Poisson processes [13]. The Poisson processes are also widely used to model technological uncertainties [13] or catastrophic events that make it impossible to proceed with the project [17]. The revenues may also be uncertain, and it may be necessary to model them with stochastic processes [16].

The real option value can be determined through closed-form valuation models, like the Geske model [14] or the Carr model [3]. In general, real options are American, which means that they may be exercised over a period of time instead of being exercised in a given moment. The value of such options is the solution of partial differential equations. The analytic solution or construction of such equations can be hard and an alternative is to use numerical techniques, analytical approximations or simulation. Notice that real options models mostly have a high complexity, because they integrate a set of interacting options, complicating their evaluation. It can be very hard to define or solve some kind of equations that

represent the real options value. Simulation is a good alternative to evaluate real options, because it allows to consider the state variables as stochastic processes and, nowadays, the simulation techniques are easy to use, transparent and flexible [10]. For example, Schwartz and Moon [16] formulate the model they use in continuous time and, then, they define a discrete time approximation and solve the model by simulation.

The model and the procedure we present intend to find the optimal strategy in terms of resource allocation to tasks of R&D projects. This optimal strategy maximizes the task value and this value characterizes and evaluates the task. In tasks of R&D projects, it is important, when different levels of resources are available, to choose the most appropriate one at each moment, that is, it is important to choose the level of resources that maximizes the task value. The evaluation is financial and to incorporate the operational flexibility and uncertainty, we use real options theory. Furthermore, we used simulation (Least Squares Monte Carlo – LSM) in the evaluation process to deal with different state variables. We elected the LSM method, because it allows making decisions according to future expectations.

The model we present is flexible enough to apply to tasks of different R&D projects. The state variables were modelled without very strong assumptions, and the necessary parameters can be inferred from the information concerning other projects of the company.

3 The Proposed Model

We present a tool that can be applied to evaluate tasks of R&D projects, in order to help management making decisions concerning resource allocation.

We consider that an R&D project is composed by different tasks, and to evaluate a project, we must evaluate its tasks. The tool herein presented intends to evaluate those tasks.

As mentioned before, the result of such evaluation is a set of rules that helps management choose the level of resources, at each moment. These rules allow maximizing the task value. To apply this procedure, it is necessary to define the finite set of levels of resources and the respective cost *per* unit of time.

We assume that each task is homogeneous and needs a certain number of identical and independent work units to be completed. These work units can be seen as small parts of the task and the set of these parts composes the task. The work units can also be executed by different resource levels, which lead to different average times to finish the task and different costs *per* time unit.

We consider, in our model, that there is uncertainty in the time it takes to complete a task, and consequently, in the costs, because they depend directly on the time to complete the task.

The costs are deterministic, *per* unit of time, and depend on the level of resources. We also assume that there may be a cost inherent to switching between different resource levels.

In the model herein presented, we do not model the revenues, but the cash flows resulting from the completion of the task. The expected operational cash flows resulting from the exploration of the investment project follow a stochastic process and depend on the time it takes to complete each task.

Notice that a set of tasks composes a project. From now on, the present value of the cash flows resulting from the lifetime of each task (cash inflows and cash outflows) is termed the task worth. These cash flows represent a portion of the total cash flows of the entire project. The concept of instantaneous task worth is used, which represents the present value of the task worth, assuming that the task was already finished. We also assume a penalty in the task worth according to its completion time, that is, the task worth is more penalized as the task takes longer to be completed. We incorporate this penalty, because we assume that R&D projects can turn more profitable if a product or a service is launched earlier. Finally, the task value is calculated through the costs, time and task worth.

Before presenting the evaluation process we define the relevant variables. Thus, the next subsections describe in detail how we handle the time to complete a task, the task worth, the costs and the net present value of the task.

3.1 Time to Complete a Task

The time to complete a task is not deterministic because it is impossible to know it with certainty, due to unpredictable delays or technical difficulties. Considering a specific level of resources along the entire task, we define the time to finish the task as $T^{(k)}$.

$T^{(k)}$ is a random variable and it is the sum of a deterministic term, the minimum time to finish the task, $M^{(k)}$, with a stochastic one. Let D be the number of work units to complete the task. The time it takes to complete each work unit is composed by a constant part and a stochastic one, the latter being defined by an exponential distribution. This distribution is adequate because we assume that the average number of work units completed *per* unit of time is constant and there is no a priori expectation as to the nature of the distribution [8]. We also assume that the time it takes to complete one work unit is independent of the time it takes to complete the other work units. Thus, it is immediate that the necessary time, $T^{(k)}$, to complete the task, using the level of resources k, is defined by

$$T^{(k)} = \sum_{i=1}^{D} \hat{t}_i^{(k)} \tag{1}$$

where $\hat{t}_i^{(k)}$ is the time that each work unit takes, considering the level of resources k. Each term can be written as

$$\hat{t}_i^{(k)} = \frac{M^{(k)}}{D} + t_i^{(k)} \tag{2}$$

where $t_i^{(k)}$ follows an exponential distribution with average $1/\mu^{(k)}$. Replacing $\hat{t}_i^{(k)}$ in (1)

$$T^{(k)} = M^{(k)} + \sum_{i=1}^{D} t_i^{(k)} \tag{3}$$

The time to finish the task is composed by a sum of a deterministic term, which represents the minimum time that is necessary to finish the task, with a stochastic one. This latter term is defined as the sum of D independent and identically distributed exponential variables.

3.2 The Costs

The costs we consider are related to the usage of the resource levels. Thus, these costs depend on the level of resources used and on the necessary time to complete the task. We assume that the costs are deterministic *per* unit of time and that they increase at a constant rate, possibly the inflation rate. Considering a specific level of resources k, let $C_x^{(k)}$ be the instantaneous cost, at instant x. The model for the costs can be defined by

$$dC_x^{(k)} = \rho C_x^{(k)} dx \tag{4}$$

where ρ is the constant rate of growth of the costs. Thus the value of $C_x^{(k)}$ is

$$C_x^{(k)} = C_0^{(k)} e^{\rho x} \tag{5}$$

where $C_0^{(k)}$ is a constant dependent of the level of resources.

The cost, $\bar{C}_j^{(k_j)}$, of a work unit j that uses the level of resources k_j, and that begins on instant x_j and ends on instant x_{j+1} is

$$\bar{C}_j^{(k_j)} = \int_{x_j}^{x_{j+1}} C_x^{(k_j)} dx = \int_{x_j}^{x_{j+1}} C_0^{(k_j)} e^{\rho x} dx = \left[\frac{1}{\rho} C_0^{(k_j)} e^{\rho x} \right]_{x_j}^{x_{j+1}} = \frac{C_0^{(k_j)}}{\rho} (e^{\rho x_{j+1}} - e^{\rho x_j}) \tag{6}$$

The present value of the cost of the work unit j, that uses the level of resources k_j, with respect to an instant x_0, and assuming a discount rate r, is $\hat{C}_{j,x_0}^{(k_j)}$ and it is given by

$$\hat{C}_{j,x_0}^{(k_j)} = \int_{x_j}^{x_{j+1}} C_x^{(k_j)} e^{-r(x-x_0)} dx = \frac{C_0^{(k_j)} e^{rx_0}}{\rho - r} (e^{(\rho - r)x_{j+1}} - e^{(\rho - r)x_j}) \tag{7}$$

The expression above is valid when $\rho \neq r$. In the case $\rho = r$,

$$\hat{C}_{j,x_0}^{(k_j)} = \int_{x_j}^{x_{j+1}} C_x^{(k_j)} e^{-r(x-x_0)} dx = C_0^{(k_j)} e^{rx_0}(x_{j+1} - x_j) \tag{8}$$

We also assume that costs related to changes of the level of resources can occur. That is, if there is a change in the level of resources from one work unit to another, it may be necessary to incur a cost. The cost for changing the level of resources from k_j, in the work unit j, to other level of resources, k_{j+1}, in the next work unit $j+1$ is given by $\gamma(k_j, k_{j+1})$. We also assume that these costs are deterministic, depend on the level of resources and grow at the same rate ρ. If the change occurs at moment x_{j+1}, that is, at the moment that work unit $j+1$ begins, the value of the respective cost is $\gamma(k_j, k_{j+1})e^{\rho x_{j+1}}$ and the present value of this cost, with respect to an instant x_0 is

$$\gamma(k_j, k_{j+1})e^{\rho x_{j+1}} e^{-r(x_{j+1}-x_0)} \tag{9}$$

In our evaluation procedure, it is necessary to calculate the present value of the total remaining costs, that is, it is necessary to determine the total costs from a certain work unit j until the last one, D. We assume that, for all work units, $j = 1, \ldots, D$, the present value of the remaining costs is determined with respect to x_j, which is the instant in which the work unit j starts, and it is denoted as $TotC(j, x_j)$. The expression of $TotC(j, x_j)$ can be given by

$$TotC(j, x_j) = \sum_{a=j}^{D-1} [\hat{C}_{a,x_j}^{(k_a)} + \gamma(k_a, k_{a+1})e^{\rho x_{a+1}} e^{-r(x_{a+1}-x_j)}] + \hat{C}_{D,x_j}^{(k_D)} \tag{10}$$

where:

- x_j is the instant in which work unit j starts;
- $\hat{C}_{a,x_j}^{(k_a)}$ is the present value of the cost of the work unit a, with respect to the instant x_j. The work unit a begins at instant x_a and uses the level of resources k_a;
- $\hat{C}_{D,x_j}^{(k_D)}$ is the present value of the cost of the work unit D with respect to instant x_j. The work unit D uses the level of resources k_D;
- $\gamma(k_a, k_{a+1})$ defines the value of the cost to change from the level of resources k_a used in work unit a to the level of resources k_{a+1} used in work unit $a+1$. Notice that if the level of resources is the same in the work unit a and in the work unit $a+1$, this cost is zero;
- r is the discount rate.

3.3 The Task Worth

Many authors define the price process of an R&D product through the geometric brownian motion (GBM), $dP = \alpha Pdt + \sigma Pdz$. The GBM assumes that uncertainties, with respect to the project, are solved along the project lifetime. This assumption can be very strong in an R&D environment, because the expected value may not be adjusted continuously. Thus, the use of a GBM would not be suitable. Notice that the expected task value may vary with shocks (positive or negative), such as the discovery of a new technology, a competitor's entry or technological difficulties, among others. These shocks occur at certain discrete moments of time, and not continuously as it is assumed when a GBM is used.

We define the task worth as the present value of the cash flows resulting from completing the task (including both cash inflows and cash outflows). The task worth does not depend on the level of resources used to undertake the task, but on the time to complete it. We also define the related concept of instantaneous task worth (or instantaneous worth), which is the value of the task worth assuming that the task is completed at the instant being considered. We assume that the instantaneous task worth changes according to a pre-defined rate and with some stochastic events. The rate can be positive or negative, depending on the nature of the project, and it can be inferred from historical data or knowledge and experience of managers. In R&D projects, new information can arrive, or unexpected events can occur, that change the course of the project and, consequently, the expectations regarding to the instantaneous task worth. Thus, we chose to model the instantaneous task worth by using a Poisson process. The parameter associated with the Poisson process is constant along the task because it is considered that, at each moment, the likelihood of a "shock" is the same. So, there is no specific information on the ongoing progress of the cash flows.

We also assume that a penalty in the instantaneous task worth may occur, due to the duration of the task. That is, we assume that it may be the case that the earlier the product is launched in the market, the bigger is the worth obtained. The task worth may be more penalized as the task takes longer to complete. The reason for such penalty may be related to the existence of competition: if a competitor is able to introduce, earlier, a similar product in the market, the task worth might be lower. Let the model of instantaneous worth, R, be defined by:

$$RdR = \alpha Rdx + Rdq \tag{11}$$

The parameter α included in the model represents the increasing or decreasing rate of the instantaneous worth, in each lapse dx. The term dq represents a Jump process, that is

$$dq = \begin{cases} u, \text{ with probability } pdx \\ 0, \text{ with probability } 1 - pdx \end{cases} \tag{12}$$

with u defined by an uniform distribution, $u \sim U(u_{min}, u_{max})$, $u_{min} \leq u_{max}$. Notice that, if the instantaneous worth would depend only on the rate α, it would be continuous and monotone increasing (assuming the rate positive). But, besides the rate α, there is the possibility of occurring jumps in the instantaneous worth, due the nature of these projects and/or the behavior of the market. New information can drastically modify the course of the project and the entry of new competitors can change the value of the project. In order to handle the model, we assume a discrete version of the instantaneous worth. Without the jump process, the solution of (11) would be $R_x = R_0 e^{\alpha x}$, and therefore we would have $R_{x+1} - R_x = R_0 e^{\alpha x}(e^{\alpha} - 1)$. Considering low values for α, we can assume that $e^{\alpha} - 1 \approx \alpha$, and the expression would become $R_{x+1} - R_x \approx \alpha R_x$. In order to incorporate the jump process, we assume that in a lapse of time that is not infinitesimal, more than one jump may take place. So, the discrete version of the model of the instantaneous worth becomes

$$R_{x+1} \approx R_x + \alpha R_x + R_x \Delta q \tag{13}$$

where $\Delta q = \sum_{i=1}^{\upsilon} u_i$, with $u_i \sim U(u_{min}, u_{max})$, and υ is defined by a Poisson distribution with parameter p, that is, $\upsilon \sim P(p)$.

The present value of the instantaneous worth in a given moment depends on the instantaneous worth of the previous moment. Thus, the instantaneous worth of the first period is $R_1 \approx R_0 + \alpha R_0 + R_0 \Delta q$. It is necessary to know the initial value R_0, which is an input parameter for the model. Assuming that the task is finished at the moment T, we define R_T as the task worth, which is calculated according to the model previously presented.

The penalty mentioned initially can be expressed by a function $g(x)$, where x denotes the time. This function is positive, decreasing, and it takes values from the interval [0, 1]. Thus, assuming this feature, the final expected task worth is $R_T \times g(T)$. Notice that, if there is no penalty, $g(x) = 1$, $\forall x$.

3.4 The Net Present Value of the Task

For the model, it is necessary to calculate the expected value of the net present value of the task, for each work unit j, $j = 1, \ldots, D$, and with respect to the initial instant of that work unit, x_j. The net present value of the task in each work unit includes the present value of the expected task worth at the end of the task and the present value of the total remaining costs. These present values are calculated with respect to the instant x_j. Thus, assuming that the instant to finalize the task is T, the net present value of the task, at the beginning of the work unit j is $Val(j, x_j)$ and it is determined as follows:

$$Val(j, x_j) = R_T \times g(T) e^{-r(T - x_j)} - TotC(j, x_j) \tag{14}$$

4 Procedure for the Task Evaluation

The procedure we propose intends to define the optimal strategy to execute an R&D task. We consider that there are different levels of resources, and the aim of this procedure is to choose which level of resources should be used at each moment, in order to maximize the task value.

The procedure considers the models presented in the previous section and uses a method similar to the Least Square Monte Carlo [10]. We selected this method for being simple, for considering different state variables and for capturing the future impact of current decisions.

The least-squares Monte Carlo (LSM) method was presented by Longstaff and Schwartz [10] and it estimates the price of an American option by stepping backward in time. At any exercise time, the holder of an American option optimally compares the payoff from immediate exercise with the expected payoff from not exercising it. This approach uses a conditional expectation estimated from regression, which is defined from paths that are simulated with the necessary state variables. The paths are simulated forward using Monte Carlo simulation, and the LSM performs backwards-style iterations where at each step it performs a least-squares approximation from the state variables [4]. The fitted value from the regression provides a direct estimate of the conditional expectation for each exercise time. Along each path, the optimal strategy can be approximated, by estimating this conditional expectation function for in-the-money paths and comparing it with the value for immediate exercise. Discounting back and averaging these values for all paths, the present value of the option is obtained.

This method can be applied to estimate the value of real options. It constructs regression functions to explain the payoffs for the continuation of an option through the values of the state variables. A set of simulated paths of the state variables is generated. With the simulated paths, the optimal decisions are set for the last period. From these decisions, it is built, for the penultimate period, a conditional function that sets the expected value taking into account the optimal decisions of the last period. With this function, optimal decisions are defined for the penultimate period. The process continues by backward induction until the first period is reached. The use of simulation allows integrating different state variables in an easy way.

The procedure herein presented is based on LSM method, but some adaptations were necessary, for example in the way time is handled.

The process consists in the following: we start by building many paths with different strategies. The strategies used to build the paths include executing all work units with the same level of resources or using different resource levels to finish the task. For each strategy, and for each path, we simulate the values of the time to execute the task, through the model we presented in the previous section. With the time and the level of resources used, we can determine the costs, and through the model for the task worth, we can simulate the values for the instantaneous worth; finally, we can determine the net present value of the task, for each path, and for each work unit.

In this procedure we build, by backward induction and for all work units, regression functions for values previously calculated for the paths. These functions explain the net present value of the task as a function of different state variables: the elapsed time, the instantaneous worth and the number of work units already finished.

The evaluation procedure begins in the last work unit. For all paths initially simulated, it is considered the instantaneous worth observed in the beginning of the last work unit, as well as the time elapsed until the start of that work unit. For all paths, assuming a specific level of resources, k_D, to complete the last work unit, the time to complete the task is redefined, as well as the net present value of the task in the beginning of the last work unit. Taking the net present value of the task recalculated in all paths, $V|k_D$, the elapsed time until the beginning of the last work unit, $Y_{1,D}$, and the instantaneous worth observed in the beginning of the last unit, $Y_{2,D}$, a regression function, F_{D,k_D}, is built. This function explains the net present value of the task, in the last unit, as function of the elapsed time until the beginning of the last work unit and of the instantaneous worth observed in the beginning of that unit. We regress $V|k_D$ on a constant, and on the variables $Y_{1,D}$, $Y_{2,D}$, $Y_{1,D}^2$, $Y_{2,D}^2$ and $Y_{2,D}Y_{1,D}$, that is,

$$F_{D,k_D} = a_0 + a_1 Y_{1,D} + a_2 Y_{2,D} + a_3 Y_{1,D}^2 + a_4 Y_{2,D}^2 + a_5 Y_{2,D} Y_{1,D} \tag{15}$$

We assumed these basis functions for the regression, but other basis functions could be selected without interfering with the process or altering it [18].

This procedure is repeated assuming the other resource levels to perform the last work unit. Thus, considering that there are N resource levels, in the last work unit, for each level of resources k_D, $k_D = 1, \ldots, N$, we define a function, F_{D,k_D}, which explains the net present value of the task as function of the elapsed time (until the beginning of the last unit) and of the instantaneous worth observed in the beginning of that unit.

For the earlier work units, the procedure is based on the same principle: it is considered that the work unit under consideration, say j, is executed with a specific level of resources, k_j. Next, the net present value of the task in the beginning of that unit is recalculated, through the definition of the best strategy from the following unit until the last one. The definition of the best strategy is done using the regression functions already determined (Fig. 1) and the costs for switching levels: for each of the following work units, the level of resources chosen is the one that leads to a higher value in the difference between the respective regression function and the cost of switching the level (if the level of resources is different from the level used in to the previous work unit), that is, for $a = j + 1, \ldots, D$, the level of resources chosen k_a is

$$\max_{k_a = 1, \ldots, N} \{F_{a,k_a} - \gamma(k_{a-1}, k_a) e^{\rho x_a}\} \tag{16}$$

Assuming the specific level of resources used in the unit j, and with the best strategy defined from the work unit $j + 1$ until the last one, we recalculate the net

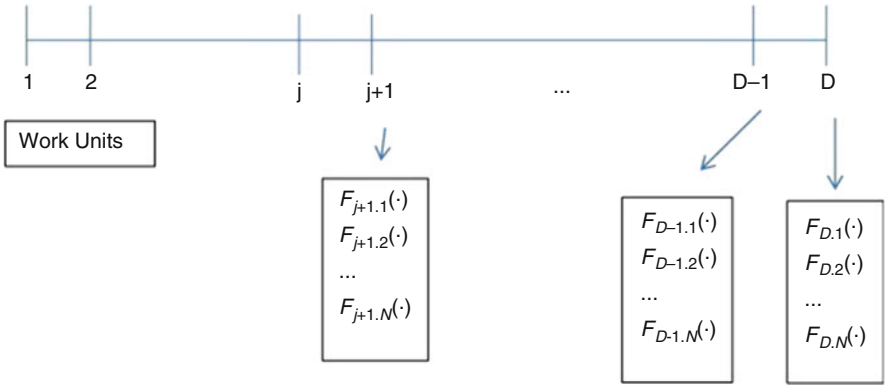


Fig. 1 Functions that allow the definition of the best strategy from unit $j + 1$ until the last unit D

present value of the task in the beginning of the unit j. Taking the recalculated values of the net present value of the task, $V|k_j$, the values of the elapsed time until the beginning of the unit j, $Y_{1,j}$, and the values of the instantaneous worth observed in the same moment, $Y_{2,j}$, a regression function is defined. This regression function explains the net present value of the task as a function of the elapsed time until the beginning of work unit j and of the instantaneous worth observed in the beginning of work unit j.

For this work unit j the procedure is repeated, assuming the other resource levels to execute it. In this way, we construct regression functions for all resource levels in the work unit j. These functions explain the net present value of the task as a function of the elapsed time and of the instantaneous worth. The process proceeds by backward induction until the second work unit. This procedure allows having, for each work unit and for all resource levels, a regression function that estimates the net present value of the task, through the elapsed time and through the instantaneous worth observed (Fig. 2).

For the first work unit we do not construct the regression functions, due the fact that, for all paths, the instantaneous worth observed in the beginning of the first unit is R_0 and the elapsed time in that moment is 0, that is, the instantaneous worth observed and the elapsed time are constant. Thus, to determine the best level of resources in the first work unit, a specific level of resources is assumed. Then, with the regression functions of the following work units, the best strategy is defined for all paths. With the best strategy in each path, the net present value of the task is calculated for the first work unit. The average of these values provides the task value, assuming that specific level of resources for the first unit. This evaluation is repeated, assuming the other resource levels for the first work unit. Notice that it is necessary to decide which level of resources may be used to begin the task. The level leading to a bigger average value of the task in the first unit is chosen to initialize the task. After this procedure, the regression functions allow defining rules which can guide management in the decisions about the strategy to use. Thus, with the

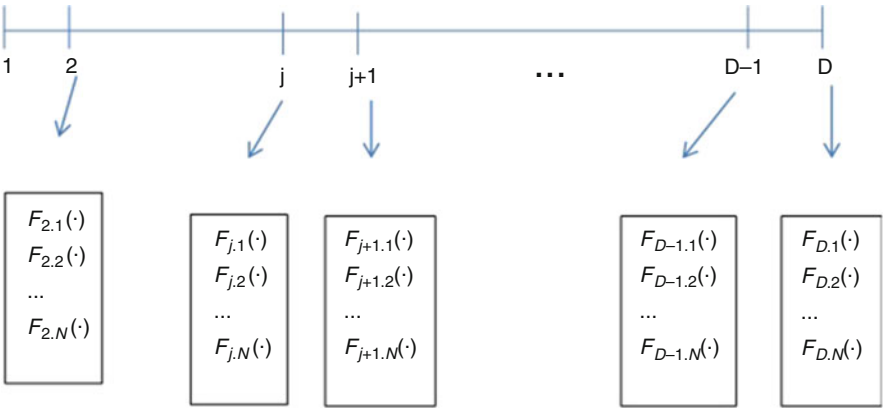


Fig. 2 Regression functions, explaining the task value as function of elapsed time and instantaneous task worth

regression functions and knowing which level of resources was used, we can define rules, indicating management which is the level of resources to use next.

5 Numerical Example

To test the evaluation procedure, we consider a project that is being executed. It is necessary to evaluate one of its tasks and define the best strategy for undertaking it. There are two different resource levels (level 1 and level 2) to execute the task. For this specific task, $D = 20$ work units were defined, that is, the task is divided in 20 identical parts. We assume that, in average, level 1 can conclude 1.5 work units *per* unit of time, and level 2 can conclude 3 work units *per* unit of time. The costs increase at a rate of 0.5 % *per* unit of time, and the discount rate is $r = 0.1\,\%$, *per* unit of time. We assume that the instantaneous task worth increases 1 % *per* unit of time. The penalty function for the instantaneous worth punishes it up to 10 %, if the task takes less than 10 units of time; if the task takes between 10 and 15 units of time, the task worth is penalized up to 30 %; if the task takes longer than 15 units of time, the penalty is fixed: 30 %. Thus, the penalty function, $g(x)$, where x represents time, is the following:

$$g(x) = \begin{cases} 1 - \dfrac{x}{10} \times 0.1, & \text{if } x \leq 10 \\ 0.9 - \dfrac{x-10}{15-10} \times 0.2, & \text{if } 10 < x \leq 15 \\ 0.7, & \text{if } x > 15 \end{cases}$$

The input parameters are in Table 1.

Table 1 Input parameters for the numerical example

Time	Costs	Task worth
$M^{(1)} = M^{(2)} = 0$	$C_0^{(1)} = 10;\ C_0^{(2)} = 40$	$R_0 = 2000$
$\mu_{(1)} = 1.5; \mu^{(2)} = 3$	$\gamma(k_j, k_{j+1}) = C_0^{(k_{j+1})}$ $\rho = 0.5\,\%$	$\alpha = 1\,\%$ $\nu \sim Po(0.4)$ $u_i \sim U(-0.2, 0.2)$

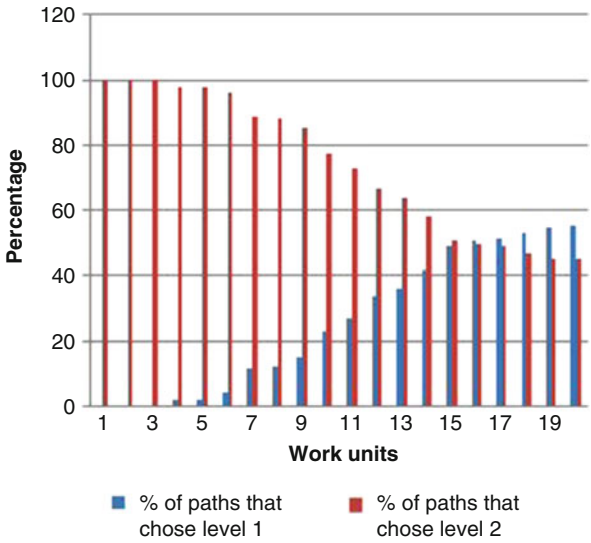


Fig. 3 Percentage of the paths that chose each level of resources, after applying the evaluation process

To run the evaluation procedure, 700 paths were considered and we used the following initial strategies: to execute the whole task with the first level of resources; to execute the whole task with the second level of resources; to execute half of the task with one level and the other half with the other level of resources. This led to a total of 2100 paths. The paths built, using only the level 1, led to an average time of 13.28, with a net present value of 1637.6. The paths built, using only the level 2, led to an average time of 6.64, with a net present value of 1708.5. After running the procedure described in the previous section, the average time to execute the task is 8.4 and the net present value of the task is 1728.58. Analyzing the results of the strategy used, level 2 is the only one chosen in the first units, but afterwards level 1 is chosen in many paths (Fig. 3).

In order to analyze the procedure, we can assay which one is the indicated level of resources for the next work unit. If we know the level of resources used before and the regression functions of the next work unit, it is possible to choose the level of resources that should execute the next work unit, taking into account the state variables information. The best decisions can be plotted as regions in the two-dimension space defined by the instantaneous worth and by the elapsed time. Such plot can provide some intuition about the best choices concerning the level of

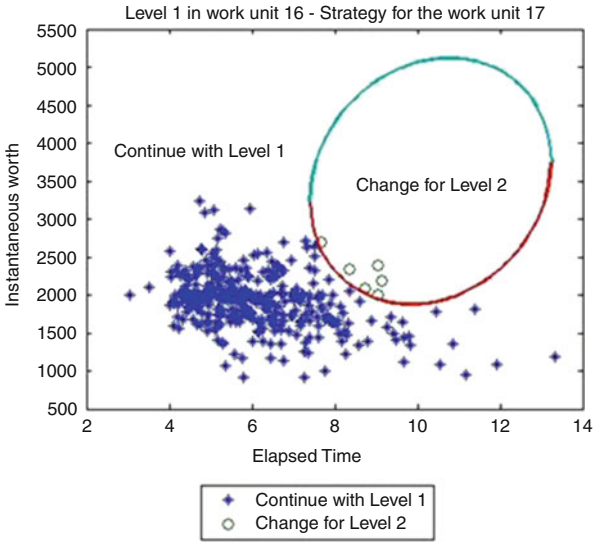


Fig. 4 Strategy for work unit 17, when level 1 is used in the work unit 16

resources to be used in the next work unit. If in a certain work unit d, level 1 was used, the equation to determine the frontier lines between "continuing with level 1" and "change to level 2" regions are is $F_{d+1,1}(\cdot) = F_{d+1,2}(\cdot) - \gamma(1,2)e^{\rho x_{d+1}}$. Similarly, if in a certain work unit d, level 2 was used, the equation to determine the frontier lines between the "continuing" and "change" regions is $F_{d+1,1}(\cdot) - \gamma(2,1)e^{\rho x_{d+1}} = F_{d+1,2}(\cdot)$.

For example, assume that unit 16 of the task is completed. Supposing that level 1 was used in unit 16, we can provide a plot that defines how the level of resources should be chosen for work unit 17. This plot defines two regions, "continue with level 1" and "change to level 2", with the frontier lines obtained through $F_{17,1}(\cdot) = F_{17,2}(\cdot) - \gamma(1,2)e^{\rho x_{17}}$. Besides these frontier lines, we also plotted the level of resources chosen for work unit 17, in the different paths in which level 1 was used in work unit 16 (Fig. 4). The little stars in Fig. 4 correspond to the paths that used level 1 in unit 16 and continue with level 1 in unit 17. The little balls correspond to the paths that used level 1 in work unit 16 and changed to level 2 in work unit 17. According to the region in which the pair (elapsed time, instantaneous worth) is situated, it is possible to define the level of resources to use in unit 17.

The procedure herein presented can be analyzed according to others aspects. We can analyze the net present value of the task when the input parameters change.

For example, we can analyze the changes when the costs to switch level exist or not; or when the penalty of the task worth exists or not. Taking the example above we obtain the following results, displayed on Table 2.

The most significant increase in the net present value of the task occurs when we remove the task worth penalty. The removal of the cost of switching level might not

Table 2 Net present value of the task considering the existence or not of a penalty in the task and/or the costs to switch level

Costs to switch level	Penalty	Net present value of the task in its beginning
Yes	Yes	1728.6
No	Yes	1733.0
No	No	2097.3
Yes	No	2097.7

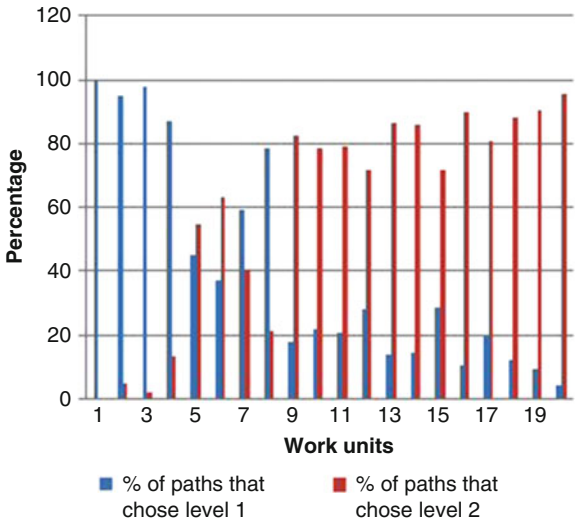


Fig. 5 Percentage of the paths that chose each level of resources, after applying the evaluation process without costs to switch level of resources

change the net present value of the task very much, but changes the strategy to use. Notice that the absence of these costs leads to more changes of level of resources, like shown in Fig. 5. This happens if there is not a dominant level, that is, if there is not a level that leads to a higher net present value of the task in all work units.

6 Usage of the Procedure

The evaluation procedure herein presented allows managing tasks of R&D projects. Knowing the levels of resources available to execute a task, the evaluation procedure provides a strategy to complete the task. The main utility of this approach is to help managers to understand what level of resources should start the task. Furthermore, analyzing the results, managers can see whether it is useful or not to change the level of resources during the task. If circumstances change, throughout the execution of the task, managers can reapply the evaluation procedure, considering a "subtask" of the initial task, that is, considering a smaller number of work units, since some

of them have been completed. In this case, managers can decide on the level of resources that should execute the rest of the task.

This evaluation process provides two useful results: the expected net present value for the task and the corresponding strategy to execute it. For the application of this approach, information is needed to allow the inference of the model parameters and is also necessary to know what levels of resources are available to execute the task. The input parameters of the evaluation model can be inferred from historical data of the company. The construction of this approach was made, taking into account it would be possible to infer these parameters. A major difficulty in applying evaluation models to real projects is the knowledge of the required parameters. Furthermore, many models consider assumptions that are difficult to be encountered in reality. This procedure tried to rely on realistic and simple assumptions.

7 Conclusions and Future Research

We developed a financial approach to evaluate homogeneous tasks of R&D projects. This approach takes into account one single criterion, which is financial; it is based on real options and its result defines the strategy to execute the tasks as well as the correspondent financial value. The strategy to execute the task consists on a set of rules that allows defining, at each moment, which is the level of resources that should be chosen, among the available levels of resources.

The resource levels impose different average speeds, as well as different costs *per* unit of time. The model incorporates the completion time of the task, the cost, the task worth and the net present value of the task. The evaluation procedure is based on a simulation process and uses, in their regression functions, information observed at each moment. If new information appears or the course of the task changes, the procedure can be reapplied. Managers can reapply the procedure whenever it is necessary.

This approach can be improved by introducing an abandonment option, when the expected net present value is equal to or lower than a certain reference value. This option must be integrated and interpreted in the context of the project that contains the task.

Considering that an R&D project is a set of interdependent tasks, this evaluation procedure can be the basis to analyze the strategy to execute an R&D project, as well as the financial value associated to it. However, there are some aspects that must be taken into account: the result of the evaluation of a task influences the evaluation of the next task. On the other hand, the connections between the tasks and the way these connections influence the evaluation of an R&D project must also be taken into account.

Acknowledgements This work has been partially supported by FCT under project grant PEst-C/EEI/UI0308/2013.

References

1. Blanco, G., Olsina, F.: Real option valuation of FACTS investments based on the least square Monte Carlo method. IEEE Trans. Power Syst. **26**(3), 1389–1398 (2011)
2. Brach, M., Paxson, D.: A gene to drug venture: Poisson options analysis. R&D Manag. **31**(2), 203–214 (2001)
3. Cassimon, D., Engelen, P.J., Yordanov, V.: Compound real option valuation with phase-specific volatility: a multi-phase mobile payments case study. Technovation **31**, 240–255 (2011)
4. Choudhurry, A., King, A., Kumar, S., Sabharwal, Y.: Optimizations in financial engineering: the least-squares Monte Carlo method of Longstaff and Schwartz. In: Parallel and Distributed Processing, IEEE International Symposium on IPDPS 2008, Miami, vol. 1, no. 11, pp. 14–18 (2008)
5. Copeland, T., Antikarov, V.: Real Options. Texere, New York (2001)
6. Cortazar, G., Schwartz, E., Casassus, J.: Optimal exploration investments under price and geological-technical uncertainty: a real options model. R&D Manag. **31**(2), 181–189 (2001)
7. Cortazar, G., Gravet, M., Urzua, J.: The valuation of multidimensional American real options using the LSM simulation method. Comput. Oper. Res. **35**, 113–129 (2008)
8. Folta, T., Miller, K.: Real options in equity partnerships. Strateg. Manag. J. **23**(1), 77–88 (2002)
9. Lint, O., Pennings, E.: An option approach to the new product development process: a case study at Philips electronics. R&D Manag. **31**(2), 163–172 (2001)
10. Longstaff, F., Schwartz, E.: Valuing American options by simulation: a simple least-square approach. Rev. Financ. Stud. **14**, 1113–1147 (2001)
11. Myers, S.: Finance theory and finance strategy. Interfaces **14**(1), 126–137 (1984)
12. Peng, H., Liu, M.: Valuation research of real options on the basis of life cycles R&D projects. In: International Conference on Management Science & Engineering (15th), Long Beach, pp. 1457–1462 (2008)
13. Pennings, E., Sereno, L.: Evaluating pharmaceutical R&D under technical and economic uncertainty. Eur. J. Oper. Res. **212**, 374–385 (2011)
14. Perlitz, M., Peske, T., Schrank, R.: Real options valuation: the new frontier in R&D project evaluation? R&D Manag. **29**(3), 255–269 (1999)
15. Santiago, L., Bifano, T.G.: Management of R&D projects under uncertainty: a multidimensional approach to managerial flexibility. IEEE Trans. Eng. Manag. **52**(2), 269–280 (2005)
16. Schwartz, E., Moon, M.: Rational pricing of internet companies. Financ. Anal. J. **56**(3), 62–75 (2000)
17. Schwartz, E., Zozaya-Gorostiza, C.: Investment under uncertainty in information technology: acquisition and development projects. Manag. Sci. **49**(1), 57–70 (2003)
18. Stentoft, L.: Convergence of the least squares Monte Carlo approach to American option valuation. Manag. Sci. **80**(9), 1193–1203 (2004)
19. Tolga, A., Kahraman, C.: Fuzzy multi-criteria evaluation of R&D projects and a fuzzy trinomial lattice approach for real options. In: Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering, Xiamen, China, pp. 418–423 (2008)
20. Yeo, K., Qiu, F.: The value of management flexibility – a real option approach to investment evaluation. Int. J. Proj. Manag. **21**, 243–250 (2003)

An Exact and a Hybrid Approach for a Machine Scheduling Problem with Job Splitting

Luís Florêncio, Carina Pimentel, and Filipe Alvelos

Abstract The unrelated parallel machine scheduling problem with job splitting and setup times is addressed in this paper.

A time-indexed integer programming formulation of the problem to minimize a weighted function of the processing occurring both before and after jobs' due dates is proposed. Moreover, we apply to a suitable decomposition of the integer programming model a recently proposed framework for decomposable integer programming/combinatorial optimization problems (SearchCol, meta-heuristic search by column generation) based on the combination of column generation and meta-heuristics.

A problem specific heuristic to use in the column generation component of the SearchCol is developed. To evaluate the effectiveness of the models and the proposed algorithms, computational tests are performed.

1 Introduction

The interest both of practitioners and researchers in studying scheduling problems exists for more than 50 years and its importance is well known among the literature, with different approaches and models developed. As industry's characteristics and demands evolved, new and different implementations on models have been proposed, with a huge variety of problem types and characteristics available today.

Scheduling plays a crucial role in today's enterprises, as appropriate timing of production is mandatory and has important financial impacts. Computational and theoretical developments have given the possibility to better accommodate the needs

L. Florêncio (✉)
Centro Algoritmi, Universidade do Minho, 4710-057 Braga, Portugal
e-mail: luisflorencio@dps.uminho.pt

C. Pimentel
GOVCOPP/DEGEI, Universidade de Aveiro, 3810-193 Aveiro, Portugal
e-mail: carina.pimentel@ua.pt

F. Alvelos
Centro Algoritmi/DPS, Universidade do Minho, 4710-057 Braga, Portugal
e-mail: falvelos@dps.uminho.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_12

of industry and complex systems can now be modelled to provide better decision making. With these developments, several realistic features have been introduced around basic concepts of production's environment and other characteristics directly related to processing, setup, sequencing of jobs, job splitting, as well as a wide range of evaluation criteria [2].

This work approaches the unrelated parallel machine scheduling problem with job splitting. In this problem a set of independent jobs must be processed on a set of unrelated parallel machines, with setup times being incurred whenever a machine switches jobs. Other system characteristics are considered such as jobs release dates, machines availability dates, preservation of machine's initial setup state and job splitting.

The job splitting property allows the split (or partition) of jobs into several lots of smaller size that can be processed in more than one machine at the same time, allowing the improvement of the scheduling plans when compared with scheduling plans where no job splitting is permitted. However, job splitting increases the complexity of the optimization models since a solution can no longer be represented solely as a completion time and a machine for each job but must be represented as a completion time and a machine *for each lot* of each job. Being so, additional decisions on how many lots for each job and their size must be taken. Job splitting differs from preemption where jobs processing being interrupted are resumed later on the same machine or another machine. In job splitting, jobs can be executed at the same time in different machines.

Processing and setup times must consider, respectively, release dates for jobs and machine's availability dates. Moreover, by preserving the initial state of each machine, job's setup can be carried from a previous scheduling plan to an actual one.

To evaluate a solution for this problem, the inventory over the planning horizon is considered. The aim is to obtain a scheduling plan that minimizes processing occurring both before and after the job's due date, motivating processing to be done the closest possible to the due date, thus avoiding unnecessary work in progress. This is achieved through penalties for processing taking place both before or after the due dates of the jobs, that increase the more distant they are from the due date and that can be adjusted, accordingly to the situation where the model is being applied and to the decision maker will. Although this objective as been seldom considered in the scheduling literature, it is of great interest in the many practical applications where work in progress stocks are undesirable.

In this paper, a new integer programming time-indexed model for the Unrelated Parallel Machine Scheduling Problem with job splitting (UPMSPjs) is proposed. The model is a compact one (has a polynomial number of variables and constraints with respect to the data of the problem) and therefore can be solved directly by a general purpose mixed integer programming solver.

A second approach, proposed in this paper, relies on a decomposition model obtained by applying a (Dantzig-Wolfe) decomposition [7] by machine to the compact model. As the subproblem of this decomposition does not have the integrality property, the lower bound provided by the linear relaxation of the

decomposition model dominates the one provided by the linear relaxation of the compact model.

The decomposition model is solved by a SearchCol algorithm. SearchCol (short for metaheuristic search by column generation) is a framework for obtaining approximate solutions to integer programming/combinatorial optimization decomposable problems [3]. As the name describes, SearchCol relies on combining Column Generation (CG) and Metaheuristics (MHs). In this work, we explore a variant of SearchCol where a general purpose mixed integer programming solver replaces the MHs.

Some specific heuristics were devised and incorporated within SearchCol for the decomposition model resolution. Although SearchCol provides exact resolution of the Subproblem (SP) that resulted from the Dantzig-Wolfe decomposition, in the context of the UPMSPjs the development of a heuristic designed around the problem's characteristics and using the dual information provided by the Restricted Master Problem (RMP) brought advantages when solving the CG. In addition, a different approach to the CG is taken, with the developed heuristic solving all SPs simultaneously. Moreover, other specific heuristics were also created to build initial solutions, to include on the first RMP, aiming to create an upper bound, to speed up the CG process and to guarantee that feasible columns are present in the RMP before starting the CG process.

SearchCol may be seen as a hybrid method combining a linear programming (to solve the RMP of CG), problem specific algorithms (to solve the SPs of CG), MHs and, possibly, a general purpose mixed integer programming solver (possibly to solve the SPs of CG when an optimal solution is desired and in the search phase replacing the MHs). Several hybrid methods have been proposed in recent years [4, 31]. SearchCol has two distinctive features: it is a general approach and relies on column generation. For a detailed description of the advantages of using decomposition methods and hybrid approaches (resulting from the combination of decomposition methods with MH) we refer to [3].

The remainder of this paper is organized as follows. Section 2 reviews with detail the literature on the parallel machine scheduling and on problems related to the UPMSPjs. Section 3 presents a compact formulation of the UPMSPjs. Section 4 presents the decomposition model to be solved by a SearchCol algorithm, which is introduced in Sect. 5 along with the specific heuristics developed. Computational experiments are presented in Sect. 6. Finally, in Sect. 7 we draw the conclusions of this work.

2 Literature Review

A significant amount of research has been done on scheduling. This section outlines machine scheduling, reviewing then parallel machine scheduling and the environment being approached in this work – unrelated parallel machine scheduling. Finally, a brief overview on hybrid methods is given.

2.1 Machine Scheduling

The study of scheduling problems goes back to the mid-1950s and since then, several works have been published on the subject [2].

For both an overview of the state of the art of scheduling problems after 1999 and a historical perspective see [2] which follows previous works by [1] and by [25]. A comprehensive survey is done on scheduling problems involving setup times or costs, classifying them according to the environment previously referred and to batching and non-batching considerations (a batch can be defined as a set of jobs to be processed as a group so setup times or costs are unique to the batch, instead of incurring a setup time/cost for each job). [37] also surveys scheduling research involving setup times. In this work, important definitions and classifications are summarized, involving job, class, sequence dependence and separability setup situations.

Unlu and Mason [32] made a comparison in order to identify – for various types of objective functions and machine environments – promising Mixed Integer Programming (MIP) formulation paradigms based on the types of decision variables such as job completion time, assignment and positional, linear ordering, time indexed and network types.

Pinedo [24] offers an exhaustive study on the scheduling problem, approaching the deterministic and stochastic models and numerous variants in each one, providing several and important definitions and classifications, as well as formulations, examples and possible approaches to solve the problems.

In the following subsections, literature on Parallel Machine Scheduling (PMS) and Unrelated Parallel Machine Scheduling Problem (UPMSP), with focus on works using job splitting properties, is reviewed, making also a brief introduction on hybrid methods and relevant literature regarding this work.

2.2 Parallel Machine Scheduling

The PMS environment is defined by [32] according to the speed of processing of the machines for the different jobs: identical machines operate at the same speed (identical machine environment: P_m); non-identical machines operate at different speeds but its speed/processing rate is consistent for all machines when processing different jobs (non-identical machine environment: Q_m). The unrelated PMS environment is, in fact, a generalization of the non-identical case as an unrelated set of parallel machines can include a set of non-identical machines [24].

Most of works on problems with job splitting properties were done for the identical PMS case. Yalaoui and Chu [36] considered the problem of identical PMS with job splitting and sequence dependent setup times to minimize maximum makespan using a heuristic method to solve it. Such method was used by [21] for the same problem, using a heuristic based on a linear programming formulation to

improve the approach of [36]. Xing and Zhang [35] also studied the job splitting property on an identical PMS problem with independent setup times to minimize the makespan, discussing cases with splitting properties and analysing a heuristic for this problem by extrapolating preemption properties.

The identical PMS case with job splitting properties was also addressed by [22, 28, 30] and [14] with the objective of minimizing total tardiness.

Shim and Kim [30] developed a Branch & Bound algorithm for the problem with independent setup times, the same due dates for all jobs and machines available from the beginning of the planning horizon, using the example of Printed Circuit Boards as an industry with these characteristics. Shim and Kim [30] stated the existence of very few research results on the parallel machine scheduling problem with job splitting property.

Kim et al. [14] approached the problem with sequence independent setup times developing a heuristic that reschedules an initial scheduling plan, by splitting jobs through rules to select jobs, subjobs and machines.

Park et al. [22] considered the problem with major/minor sequence dependent setup times embedding a heuristic that accounts for the problem's properties (job splitting, setup dependency) in three existing algorithms, and comparing them to the original ones.

Sarıçiçek and Çelik [28] proposed both a Tabu Search and Simulated Annealing meta-heuristic for the problem with independent setup times and developing a MIP formulation with positional variables, finding that the Simulated Annealing approach significantly outperforms the Tabu Search in computational time and optimal solution deviation.

The objective functions considered in the reviewed literature have no resemblance to the one being studied in this work. Most works rely on evaluating scheduling plans through earliness and/or tardiness, makespan or other factors related to setup, completion times or due dates fulfilment.

As pointed out by [35] and [38], the *NP*-hardness of the problem of scheduling n jobs on m machines with distinct release dates for jobs and machines, and distinct due dates for jobs, implies that alternatives to exact approaches must be sought.

2.3 Unrelated Parallel Machine Scheduling

A survey of the literature focusing on the UPMSP without side conditions was done by [23]. The authors reviewed the several performance evaluation methods and compiled existing algorithms for the various objective functions. Pfund [23] also report that unrelated PMS environments remained relatively unstudied, noting that there were few solution approaches to minimize due date related functions, and making aware that research in this area should include the development of solution algorithms to minimize due date related criteria.

Logendran and Subur [18] studied the UPMSP, with job splitting and distinct release dates for jobs and machines, to minimize total weighted tardiness. The

authors present a MIP model using assignment and positional decision variables. In this work, the splitting property considers a job can only be split in two parts to prevent higher Work In Progress (WIP), with a predetermined number of jobs to be split. Also, neither setup times or costs are explicit, assuming they are included in the processing times Though the authors study an unrelated case with job splitting, the presented model constraints force jobs to be processed in the same machine in case a splitting occurs. To solve the problem, different initial solutions are created and then used by a Tabu Search based heuristic to find a better solution, comparing then the initial solutions that provide better results after applying Tabu Search. Logendran et al. [19] studied a similar problem with a similar approach, considering six different Tabu Search algorithms and four different initial solution methods that act as seeds of the algorithms. This work does not consider the possibility to split jobs. Sequence dependent setup times, as well as distinct release dates, with the objective of minimizing total weighted tardiness define the main characteristics of this work.

Zhu and Heady [38] developed a MIP for the Earliness-Tardiness case of an unrelated PMS problem with sequence dependent setups to provide optimal solutions for small scale problems regarding future research and validation on industrial-scale heuristics.

Shim and Kim [29] considered the problem of scheduling jobs on unrelated PMS to minimize total tardiness without setup considerations, using a Branch & Bound algorithm approach with several developed dominance rules.

Liaw et al. [16] also considered the problem of unrelated PMS to minimize the total weighted tardiness without setup considerations. They first created upper and lower bounds, through a two-phase heuristic and an assignment approach respectively, and use a Branch & Bound algorithm with dominance rules to eliminate unpromising partial solutions.

Chen and Wu [6] presented a heuristic combining the Threshold-Accepting method with Tabu Search and designed improvement procedures to minimize total tardiness for an UPMSP with auxiliary equipment constraints. Chen [5] combined the Simulated Annealing method, apparent tardiness cost with setup and designed improvement procedures to minimize total tardiness for an UPMSP with setup times that are dependent both on job sequence and machine used.

Wang et al. [34] modeled the PMS problem with job splitting, for both identical and unrelated cases and without setup considerations, to minimize the makespan, approaching it through a hybrid Differential Evolution method.

Vallada and Ruiz [33] developed a MIP with positional variables and proposed a genetic algorithm approach for the UPMSP to minimize the maximum makespan of a scheduling plan with sequence dependent setup times for both jobs and machines.

Rocha et al. [26] considered the problem of unrelated PMS, with sequence dependent setup times for both machines and jobs, and developed a Branch & Bound algorithm in order to minimize the maximum makespan and the total weighted tardiness (both are added in the same objective function).

Kim et al. [13] presented a Simulated Annealing approach for the UPMSP with job sequence dependent setup times to minimize total maximum tardiness.

A particularity in this work is the existence of already defined and divided parts of jobs.

Lopes and Carvalho [20] studied the UPMSP with sequence dependent setup times to minimize total weighted tardiness, using a Branch & Price algorithm with a new column generation acceleration method that significantly reduces the number of explored nodes.

Fanjul-Peyro and Ruiz [10] approached the UPMSP under makespan minimization. The unrelated environment was also extended to consider only a subset of desirable machines and jobs in order to understand if production capacity needs to be increased. Three algorithms were developed, combining them with CPLEX [12] or between them.

Lee et al. [15] suggested a Tabu Search algorithm to solve the unrelated PMS problem with sequence and machine dependent setups to minimize total tardiness. The Tabu Search approach outperformed significantly an existing Simulated Annealing method, and gave quicker solutions than an iterated greedy solution although it did not improve the solution values.

Lin et al. [17] approached the UPMSP using different heuristics and a genetic algorithm. In this work, neither setup times nor job splitting properties are considered. Rodriguez et al. [27] also approached the unrelated environment without setup times and splitting properties to minimize the total weighted completion times but using an iterated greedy algorithm to solve large-scale size instances.

Most research on machine scheduling problems has been done for the single machine case [2, 38]. According to the review in [2], there has been few attempts at tackling the problem of optimizing the scheduling and splitting of jobs subject to release dates and sequence-independent setup times in an unrelated parallel machines environment. Moreover, no work has been found for the particular case of using all of the referred properties being studied in this work in the same UPMSP.

3 Compact Model

The proposed compact model to the UPMSPjs is a time indexed one. In a time-indexed formulation, time is divided into a pre-set of identical periods of a unit length. The notation used in this paper is presented below.

The sets considered are represented by:

J – Set of jobs, indexed by $j = 1 \ldots n$
T – Set of discrete, integer time periods, indexed by $t = 0 \ldots T_{max}$
M – Set of machines, indexed by $i = 1 \ldots m$

The parameters are the following:

p_{ij} – Processing time of job j in machine i, in time units
r_j – Release date of job j, the moment in time it becomes available for processing
q_i – Release date of machine i, the moment in time after which machine i can process jobs

d_j – Due date of job j
w_j – Priority or weight of job j
s_j – Setup time of job j
$z[i]$ – job programmed for machine i at the beginning of the scheduling horizon
β – Constant between 0 and 1

The model's decision variables are:

$$x_{ijt} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine } i \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ijt} = \begin{cases} 1 & \text{if a setup for job } j \text{ is incurred in machine } i \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$e_{it} = \begin{cases} 1 & \text{if the setup status of machine } i \text{ changes from } z[i] \text{ to } j \text{ in period} t \\ & \text{or in a previous period of time} \\ 0 & \text{otherwise} \end{cases}$$

The developed MIP model, considering the previous presented notations and decision variables is the following:

$$Min\ Z = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{t=1}^{t\le d_j}(1-\beta)(d_j-t)w_j x_{ijt} + \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{t>d_j}^{T_{max}}\beta(t-d_j)w_j x_{ijt} \tag{1}$$

Subject to:

$$\sum_{i=1}^{m}\sum_{t>max\{r_j,q_i\}}^{T_{max}}\frac{1}{p_{ij}}x_{ijt} \ge 1 \qquad \forall j \tag{2}$$

$$\sum_{k=t-s_j}^{t-1} y_{ijk} \ge (x_{ijt} - x_{ij(t-1)})s_j \qquad \forall i, \forall j : j \ne z[i], \forall t : t \ge 1 \tag{3}$$

$$s_{z[i]}(1-e_{i(t-1)}) + \sum_{k=t-s_{z[i]}}^{t-1} y_{iz[i]k} \ge (x_{iz[i]t} - x_{iz[i](t-1)})s_{z[i]} \qquad \forall i, \forall t : t \ge 1 \tag{4}$$

$$\sum_{j=1:j\ne z[i]}^{n}\sum_{k=0}^{t}(y_{ijk} + x_{ijk}) \le te_{it} \qquad \forall i, \forall t : t \ge max\{q_i, 1\} \tag{5}$$

$$\sum_{j=1}^{n} x_{ijt} + \sum_{j=1}^{n} y_{ijt} \le 1 \qquad \forall i, \forall t : t \ge 1 \tag{6}$$

$$x_{ijt} = 0 \qquad \forall i, \forall j, \forall t : t \le max\{r_j, q_i\} \tag{7}$$

$$y_{ijt} = 0 \qquad \forall i, \forall j, \forall t : t \leq q_i \tag{8}$$

$$x_{ijt}, y_{ijt}, e_{it} \in \{0, 1\} \tag{9}$$

The objective function penalizes early and tardy processing (related to the due date), with the possibility to define if the penalization should be bigger for earlier or tardier processing, by using a convex combination with weight β. Also, within the set of processing periods before and after the due date, the more distant the processing is done from the due date the more it penalizes the solution's value. In addition, the priority of the job is also taken into account.

A set of constraints must be defined to guarantee the satisfaction of demand, considering that processing can be executed in any machine with different relations between the speed/processing and the total demand or needed processing. Constraints (2) simply state that the sum of the processing for each job must be at least equal to its total processing time (each job is completely executed and demand is satisfied).

A second set of constraints in this model relates to setup considerations, guaranteeing not only the mandatory machine setup before any new job is processed, but also the preservation of an initial setup state inherited from the previous scheduling plan. Constraints (3) ensure that a setup time is incurred whenever a machine starts processing a new job. Constraint set (4) has the same role as (3), but considers the case where the incoming job is the preprogrammed one, and allows for initial setup preservation through the change of status variable e_{it}. By (5), the setup status of a machine changes in time t if the incoming job is not the preprogrammed one (in which case a setup must be incurred before any processing takes place).

A third set of constraints aims at limiting the status of a machine, if not idle or unavailable, to one of the two active possible states: being setup or processing. By (6), it is ensured that at any given time t, a machine is either processing, being setup for a job, or idle.

A set of constraints must also be considered to guarantee that the release dates are respected, so that machines cannot process or be setup before being available and jobs cannot be processed before being also available. Constraints (7) guarantees that no processing takes place before the maximum between the release dates of job j and machine i. By (8) it is stated that setups cannot take place before machine i is available. Finally, constraints (9) bound the variables of the problem.

In Fig. 1, an example of a complete scheduling plan is provided, highlighting the benefits of applying a job splitting property in this problem.

4 Decomposition Model

In this section, a Dantzig-Wolfe decomposition [7] is applied to the compact model of Sect. 3, and the resulting Master Problem (MP) and set of smaller and independent problems – the Subproblems (SPs) – are presented.

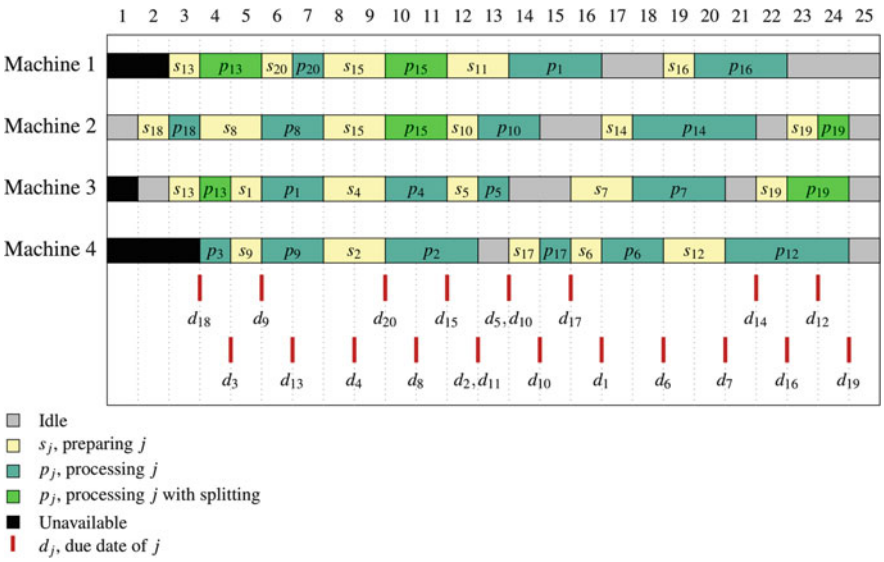


Fig. 1 A scheduling plan example

A natural decomposition of the problem is to define subproblems by machine as most constraints imposed on a single machine. Using the previously presented compact model as the original model, constraints (2) are the only where variables associated with different machines appear in the same constraint, therefore they are treated as coupling constraints. Constraints (3), (4), (5), (6), (7), (8), and (9) define the SPs. In this decomposition, each SP solution corresponds to a machine schedule, so each problem represents a single machine scheduling problem.

The SPs solutions will result in different machine scheduling plans where not all jobs must be scheduled (no satisfaction of the demand must be guaranteed) and where allocation of processing and of setup times has exactly the same procedure as in the compact model, indicating for that SP which jobs or parts of jobs (and in which periods) shall be processed. Each SP has its own characteristics and depend on machine properties and its job processing times, resulting in a set of different (sub-)problems.

For the MP new decision variables will be needed, to represent the extreme points generated by the SPs. The solution of the MP will represent a convex combination of these points.

All notation being used in this section has already been introduced in Sect. 3, except for the set, decision variables and parameters that will be presented in the following paragraphs.

A new set must be defined representing the total number of scheduling plans generated by the SPs:

H_i – Set of machine i scheduling plans, indexed by $h = 1 \ldots g_i$

The new decision variables to be used are the following:

$$\lambda_i^h - \text{Weight of scheduling plan } h \text{ of machine } i$$

A new set of parameters to be used must be defined such that:

$$\alpha_{ijt}^h = \begin{cases} 1 & \text{if job } j \text{ is processed in machine } i \text{ in period } t \text{ on scheduling plan } h \\ 0 & \text{otherwise} \end{cases}$$

It must be noted that the parameter α_{ijt}^h is directly related to the previously defined x_{ijt}, though this one is now used in the MP whereas the original one is being used by the SPs.

4.1 Master Problem

With all the notation developed in Sects. 3 and 4 the following MP can be defined:

$$\begin{aligned} Min\, Z^{MP} = & \sum_{h=1}^{g_i} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{t=1}^{t \le d_j} \left((1-\beta)(d_j - t) w_j \alpha_{ijt}^h \right) \lambda_i^h + \\ & \sum_{h=1}^{g_i} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{t>d_j}^{T_{max}} \left(\beta (t - d_j) w_j \alpha_{ijt}^h \right) \lambda_i^h \end{aligned} \tag{10}$$

Subject to:

$$\sum_{h=1}^{g_i} \lambda_i^h = 1 \qquad \forall i \qquad (\eta_i) \tag{11}$$

$$\sum_{h=1}^{g_i} \sum_{i=1}^{m} \Big(\sum_{t>max\{r_j, q_i\}}^{T_{max}} \frac{1}{p_{ij}} \alpha_{ijt}^h \Big) \lambda_i^h \ge 1 \qquad \forall j \qquad (\Pi_j) \tag{12}$$

$$\lambda_i^h \in \{0, 1\} \tag{13}$$

The objective function in (10) follows the one used in the compact model, minimizing processing occurring distant from the due date of the jobs in the chosen scheduling plans. A new set of constraints is introduced. The set of constraints (11) are the convexity constraints of the model, that guarantee that a combination of the SPs is chosen. Constraints (12) derive from the compact model set of constraints that ensure processing and satisfaction's demand is met for all jobs (in all chosen

plans for all machines). The last set of constraints, the set (13), defines the decision variables domain.

A modified version of the Master Problem is considered during the decomposition model resolution process – Restricted Master Problem (RMP), which works only with a sufficiently meaningful subset of variables [9]. When solving the linear relaxation of the RMP, a set of dual variables is obtained: Π_j from (12) giving information of whether it is attractive to process job j and from (11) the convexity constraint dual variable η_i.

4.2 Subproblem

Using the constraints sets (3), (4), (5), (6), (7), (8), and (9) of the compact model and the dual variables provided by the MP, the following SP is obtained, for machine i:

$$\begin{aligned} Min\, Z^{SP_i} = & \sum_{j=1}^{n} \sum_{t>d_j \wedge t>max\{r_j,q_i\}}^{Tmax} \left(\beta(t-d_j)w_j - \frac{1}{p_{ij}}\Pi_j\right) x_{ijt} + \\ & \sum_{j=1}^{n} \sum_{t\leq d_j \wedge t>max\{r_j,q_i\}}^{Tmax} \left((1-\beta)(d_j-t)w_j - \frac{1}{p_{ij}}\Pi_j\right) x_{ijt} + \\ & \sum_{j=1}^{n} \sum_{t\leq d_j \wedge t\leq max\{r_j,q_i\}}^{Tmax} (1-\beta)(d_j-t)w_j x_{ijt} + \\ & \sum_{j=1}^{n} \sum_{t>d_j \wedge t\leq max\{r_j,q_i\}}^{Tmax} \beta(t-d_j)w_j x_{ijt} - \eta_i \end{aligned} \tag{14}$$

Subject to:

$$\sum_{k=t-s_j}^{t-1} y_{ijk} \geq (x_{ijt} - x_{ij(t-1)})s_j \qquad \forall i, \forall j : j \neq z[i], \forall t : t \geq 1 \tag{15}$$

$$s_{z[i]}(1 - e_{i(t-1)}) + \sum_{k=t-s_{z[i]}}^{t-1} y_{iz[i]k} \geq (x_{iz[i]t} - x_{iz[i](t-1)})s_{z[i]} \qquad \forall i, \forall t : t \geq 1 \tag{16}$$

$$\sum_{j=1:j\neq z[i]}^{n} \sum_{k=0}^{t} (y_{ijk} + x_{ijk}) \leq te_{it} \qquad \forall i, \forall t : t \geq max\{q_i, 1\} \tag{17}$$

$$\sum_{j=1}^{n} x_{ijt} + \sum_{j=1}^{n} y_{ijt} \leq 1 \qquad \forall i, \forall t : t \geq 1 \tag{18}$$

$$x_{ijt} = 0 \qquad \forall i, \forall j, \forall t : t \leq max\{r_j, q_i\} \tag{19}$$

$$y_{ijt} = 0 \qquad \forall i, \forall j, \forall t : t \leq q_i \tag{20}$$

$$x_{ijt}, y_{ijt}, e_{it} \in \{0, 1\} \tag{21}$$

In this formulation, each SP is associated to a machine upon which the reduced cost of scheduling plans are evaluated at each iteration of the CG algorithm.

The set of constraints of the SP has the same meaning as in the case of the compact model (see Sect. 3).

The linear relaxation of the decomposition model is solved by CG. CG is an iterative process, typically used to solve large linear programming problems to obtain good lower bounds for integer programming problems, where the MP and the SPs interchange information between each other in each iteration of the CG process. When the RMP is solved, whether in the first iteration or in the successive iterations, it provides dual information that is included in the subproblems objective function.

Moreover, the solutions resulting from solving the SPs, in the first iteration of the CG process and the remaining ones, are iteratively added to the RMP as variables, until in a given iteration all the SPs solutions have non-negative reduced cost, meaning that no attractive columns were found on that iteration. As a result, no column will be added to the RMP in that iteration and it can be concluded that the solution of the last solved RMP is the optimal one. For a detailed description of the CG algorithm see [8].

5 SearchCol

SearchCol [3] will be used to solve the aforementioned machine scheduling problem. As the full name of the SearchCol framework suggests ('metaheuristic search by column generation'), SearchCol is a framework for combining Column Generation and metaheuristics. SearchCol can be easily extended to accommodate the use of a general purpose mixed integer programming solver. Using the SearchCol framework – and in particular a decomposition model – is attractive due to the complexity of the problem we want to study, where larger and more realistic instances are too hard to solve by non-decomposable models.

The SearchCol's global algorithm can be divided in three main steps or phases, as represented in Fig. 2. The SearchCol steps can be identified as the CG, the Search and the Perturbation phases. In each one of these steps, several methods and possible problem specific implementations are defined by the framework.

The SearchCol algorithm starts by applying CG using the subproblem heuristic detailed in Sect. 5.2. In this step the decomposition model is solved and an optimal Linear Relaxation (LR) solution to the problem is obtained, which also provides

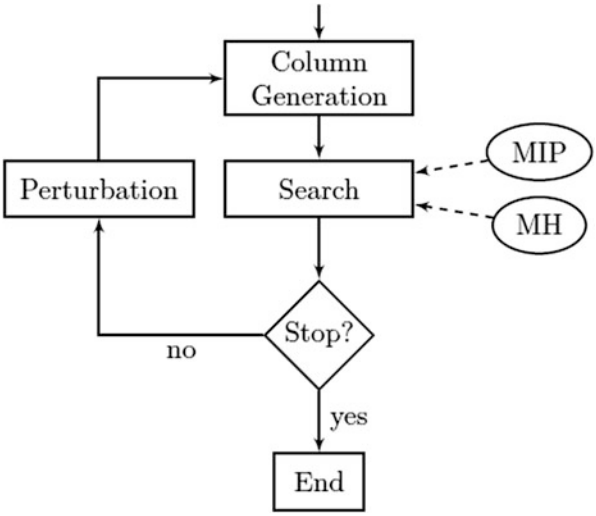


Fig. 2 SearchCol flowchart

a lower bound to the optimal integer solution. Problem specific algorithms can be implemented to solve the SPs more efficiently.

In the second phase of the algorithm, the set of previously generated, by CG, SP solutions define a search space, and work as components of the overall solution. The objective is to obtain integer solutions for the problem, that is, with $\lambda_i^h \in \{0, 1\}$, implying a column must be chosen for each SP (or machine scheduling plan). This phase is problem independent and can be conducted by a general purpose MIP solver or by a MH.

In the third step a set of perturbations is added to the RMP that will be solved in the following CG. Each perturbation is a new constraint forcing one SPs variable to take value 0 or 1 (depending on the perturbation definition). Perturbations are included with the purpose of leading CG to generate new SP solutions. The perturbation phase and its components are presented and detailed in Sect. 5.3.

After the perturbation phase, a new CG is run – perturbed CG – which will lead to a new search phase. Afterwards, if no stopping criteria is met, new perturbations are applied and the process iterates again. In the SearchCol framework, several problem independent perturbations are available, based on information such as the incumbent and/or the optimal LR solution and with deterministic or probabilistic characteristics.

The stopping criterion in SearchCol can be met using: a time limit, a limited number of search iterations, a given improvement on the value of the incumbent solution or a limited number of total iterations without improvement (a total iteration comprises the execution of the three referred steps). A more detailed description on SearchCol is available in [3].

In the following subsections we will present the developed heuristics in this work, namely heuristics for the generation of initial solutions to be included in the first RMP before the first iteration of the CG algorithm and a heuristic to solve the SPs.

5.1 Initial Solutions

The initial solutions are obtained from five different types of heuristics that create a scheduling plan for each machine with the guarantee that all jobs are fully processed in the set of all machines scheduling plans.

For each resulting machine scheduling plan, an associated variable (λ_i^h) is inserted in the RMP.

The scheduling plans are created using an average processing time rule or a due date rule to allocate all parts of each job to the most attractive periods of a machine, that is chosen from the set of all machines, after calculating which one induces the minimum weight on the objective function if processing is done on the available periods before the due date (or just after the due date if early processing is possible).

For detailed information about the developed initial solutions refer to [11].

5.2 Global Subproblem Heuristic

In SearchCol, it is possible to solve SPs heuristically or exactly, and in the former case, to use independent heuristics (each SP is solved independently) or global heuristics (a global solution that considers all SPs is obtained heuristically and then decomposed in one solution for each SPs).

The following global heuristic was devised considering the complete set of machines (SPs), meaning that, unlike common CG procedures, all SPs are solved at once.

The motivation for using this approach was to guarantee that, in an integer solution to the decomposition model, different feasible solutions are obtained. When the SPs are solved independently, because of the type of dual information provided to the SP and although the optimal solution of the LR of the master problem is feasible, forcing the selection of only one schedule plan for each machine (during the search phase) may result on an integer solution equal to the best initial heuristic solution.

As mentioned, this is caused by the type of dual information given by the RMP, as it provides the SPs with the same dual information (variable π_j) for all periods of a given job, with the reduced cost for each machine (and its associated SP) differing only because of the different processing times for the same job on different machines, which may be insufficient for an overall solution when building the solution of each machine. This aspect results in the achievement of very similar schedules for different machines in terms of the jobs considered, in a given iteration

of the CG when the SPs are being solved independently. Moreover, the most attractive jobs have a high probability of being scheduled in all machines (in a given iteration of the CG) and other jobs have a high probability of not being scheduled on any machine, resulting in overall poor integer solutions.

The global heuristic defines the schedule plans for all machines at one step in an iteration of CG, providing the RMP with all SPs solutions that can be easily seen as a global feasible solution (given the heuristic assures that all the jobs are scheduled completely). In the following paragraphs we describe the heuristic's steps. An explanation is given after to the meaning of 'cost' in the heuristic context.

1. Sort jobs hierarchically:
   (a) Increasingly by the most negative 'cost' from the set of the job's periods;
   (b) Increasingly by due date;
   (c) Decreasingly by weight;
   (d) Increasingly by index.
2. On sorted list, pick the first job not totally scheduled.
3. Sort machines increasingly by 'cost' of processing in the available periods.
4. Schedule job selected on Step 2 taking into account the first machine of the sorted list of machines from Step 3, and on first available period with the most negative cost on the objective function.
   (a) If no available period, change to the next machine on the sorted list;
   (b) If no available period and all machines checked, restart from Step 4 allowing processing on periods with non negative cost (only for the job being scheduled).
5. Repeat the process starting from Step 2 until all jobs are scheduled.

The 'cost' of processing a given job j in a given period t (as referred in Step 1) is calculated using the SPs objective function for the period being considered as shown in Eq. (22). In Step 3 the calculation is similar, although a set of given periods needed to totally process the job is considered, with the 'cost' of processing in the periods being the sum of 'costs' of each period in the interval, guaranteeing processing is possible.

$$cost = \begin{cases} \beta(t - d_j)w_j - \frac{1}{p_{ij}}\Pi_j & \text{if } t > d_j \wedge t > max\{r_j, q_i\} \\ (1 - \beta)(d_j - t)w_j - \frac{1}{p_{ij}}\Pi_j & \text{if } t \leq d_j \wedge t > max\{r_j, q_i\} \end{cases} \tag{22}$$

5.3 Perturbations

A perturbation in SearchCol is an additional constraint, i.e. a constraint not present in the original decomposition model, that fixes SP variables to 0 or to 1. In the particular case of the UPMSPjs, a perturbation forces a given period, t, of a given

machine, i, to be occupied by a given job, j, by fixing $x_{ijt} = 1$ or forbids a given job, j, to occupy a given period, t, of a given machine, i, by fixing $x_{ijt} = 0$.

The SearchCol framework states several ways of defining sets of perturbations. In the SearchCol algorithm implemented in this work, perturbations are based on a combination of the incumbent solution and the (optimal) solution obtained the last time CG was solved.

A user defined parameter (θ) is first applied to round the fractional values, from which results the values to be combined with the incumbent solution (x'_{LR_ijt}).

$$x_{LR\_ijt} \leq \theta \rightarrow x'_{LR\_ijt} = 0$$

$$x_{LR\_ijt} > 1 - \theta \rightarrow x'_{LR\_ijt} = 1$$

The following rules are then applied:

1. Subproblem variables with value 1 in the optimal LR solution and value 1 in the incumbent solution are fixed to 1;
2. Subproblem variables with a fractional value in the optimal LR solution and value 0 in the incumbent solution are fixed to 0.

Further details on perturbations are available in [3, 11].

6 Computational Tests

The implementation of the SearchCol algorithm to the UPMSPjs relied on SearchCol++ (a computational framework in C++ for the implementation of SearchCol algorithms, http://searchcol.dps.uminho.pt/). SearchCol++ uses the CPLEX callable libraries for the solution of the RMP, as a general purpose mixed integer programming solver when required for the search phase (alternatives are VNS and tabu search among other MHs) and also to solve the compact model.

Problem specific classes implementations were coded using the integrated development environment Microsoft Visual C++ in Microsoft Visual Studio 2010 with CPLEX 12.2 libraries [12] for a *x64* platform.

The problem instances were adapted from [20] and are composed of 16 subsets classified by number of jobs and machines as can be seen in Table 1. Each subset contains 5 instances corresponding each to different scheduling congestion levels (q), with the total number of instances amounting to 80.

Table 1 Instances' characteristics

M (machines)	2	2	2	2	4	4	4	4	6	6	6	8	8	8	10	10
J (jobs)	20	30	40	50	30	40	50	60	40	50	70	40	60	80	70	100

In [20], the setup times are sequence dependent whereas in this work they are sequence independent. These test instances embody all the remaining characteristics of the UPMSPjs being studied, except for the setup times which were adapted, by calculating the average setup time for all the possible sequences of any job j.

An important characteristic to consider in these instances is their scheduling system congestion level (q). For each pair of instances characteristics (machines and jobs – in Table 1) there are five different levels of congestions. The authors considered that the larger the value q the more congested the system will be. This parameter has particular importance not only because it helps to define the values of the due dates for each instance [20], but also because it causes a greater number of tardy jobs, as it becomes impossible (in more congested instances) to allocate all jobs before their respective due date.

The β parameter of the objective function was set to 0.99 and the number of periods (set T) for each instance, is calculated by:

$$T_{max} = \max \left\{ \max_j d_j + 1 \ , \ \frac{\max_j r_j + \max_i \left[\sum_j \left(p_{ij} + s_j \times m \right) \right]}{m} \right\} \tag{23}$$

All tests were run with a time limit of 1800 seconds on a PC Intel Core i7 3610QM 2.3 GHz and 8 GB RAM under MS Windows 7 x64.

Despite the referred time limit, a second stopping criterion was used so that the algorithm would stop if the incumbent solution's value was not improved after the search step.

In Tables 2 and 3 the results from the computational tests using all the heuristics previously presented (for initial solutions and solving the subproblem globally), the general purpose mixed integer programming solver in the search step of SearchCol, and the perturbations described in Sect. 5.3, are presented. The compact model results, obtained by solving it directly with CPLEX 12.2, are represented in columns denominated by 'Comp' and the SearchCol algorithm results are represented by 'Dec'. Each line corresponds to a pair of machines and jobs (M-J) and contains five instances values, one for each congestion level column represented by 'q'.

Comparing solution values, the compact model solved directly by a state-of-the-art commercial general purpose mixed integer programming solver outperforms the decomposition model solved by SearchCol in almost all instances.

For the instances with more machines and jobs and high congestion, SearchCol is able to provide a feasible solution while the compact model is not (for these instances the solution value is substituted for '*inf*').

Moreover, regarding computational times, when considering low congested instances the compact model is more efficient that the decomposition approach, whereas for higher congestion levels it requires longer computational times than the decomposition model.

Table 2 Comparison of values between models for the five congestion levels

	$q = 1$		$q = 2$		$q = 3$		$q = 4$		$q = 5$	
M-J	Comp	Dec	Comp	Dec	Comp	Dec	Comp	Dec	Comp	Dec
2–20	0.2	0.2	0.7	0.8	14.8	79.6	88.7	142.5	146.3	219.4
2–30	0.7	0.8	4.1	5.4	171.0	276.6	367.2	491.6	521.3	652.1
2–40	0.6	0.8	4.4	6.4	189.2	324.3	515.8	706.3	764.1	1088.1
2–50	0.6	1.2	1.4	2.9	193.2	449.1	755.2	1059.0	1246.4	1511.4
4–30	0.1	0.1	6.2	23.4	94.3	145.7	166.4	208.0	231.7	269.3
4–40	0.1	0.1	9.2	38.0	90.6	141.3	198.3	267.8	290.2	372.5
4–50	0.1	0.2	1.5	3.1	33.9	69.0	170.2	249.8	319.4	450.5
6–60	0.2	0.3	18.0	35.1	111.6	250.5	416.3	639.3	644.9	838.1
6–40	0.0	0.0	0.1	0.1	19.2	77.8	100.4	153.9	163.6	214.2
6–50	0.1	0.1	5.2	18.4	77.6	170.9	232.8	293.4	347.6	406.1
6–70	0.0	0.1	7.1	21.5	68.6	146.5	270.4	403.2	462.4	582.2
8–40	0.0	0.0	4.0	6.0	37.7	83.2	94.1	120.9	116.8	162.4
8–60	0.0	0.0	18.9	35.0	107.3	168.3	224.8	305.1	317.0	382.3
8–80	0.0	0.0	3.0	7.6	20.1	128.0	248.1	389.5	471.4	555.0
10–70	0.0	0.0	11.0	27.9	88.3	174.2	218.0	350.8	317.8	433.8
10–100	0.0	0.0	0.1	7.6	69.7	119.8	inf	558.3	inf	757.0
Average	0.2	0.2	5.9	14.9	86.7	175.3	271.1[a]	385.4[a]	424.1[a]	542.5[a]

[a] The average value for these columns excludes instances 10-100-4 and 10-100-5 (M-J-q)

Table 3 Comparison of time spent between models (in seconds) for the five congestion levels

	$q = 1$		$q = 2$		$q = 3$		$q = 4$		$q = 5$	
M-J	Comp	Dec	Comp	Dec	Comp	Dec	Comp	Dec	Comp	Dec
2–20	1	832	1	163	2	133	8	77	8	95
2–30	3	1814	6	1825	1674	1336	1800	306	1800	231
2–40	7	1836	7	1804	1800	1804	1800	1036	1800	618
2–50	10	1800	8	1808	1800	1806	1800	1808	1800	1807
4–30	1	182	2	582	134	218	83	129	140	128
4–40	2	70	11	875	1800	830	1350	560	1641	515
4–50	5	1809	11	1818	1438	1808	1800	1399	1800	1147
6–60	11	1809	21	1837	1800	1807	1800	1811	1800	1815
6–40	3	13	4	762	125	418	471	334	355	302
6–50	5	911	12	1146	1800	1338	1800	716	1800	660
6–70	15	1820	29	1831	1800	1823	1800	1824	1800	1814
8–40	3	9	4	462	49	368	124	205	29	204
8–60	9	428	37	1822	1640	1813	1800	1279	1800	991
8–80	28	74	32	1872	1370	1809	1800	1825	1800	1837
10–70	19	163	44	1817	1800	1835	1800	1830	1800	1651
10–100	77	570	89	1844	1800	1841	1800	1810	1801	1803
Average	13	884	20	1392	1302	1312	1365	1059	1373	976

7 Conclusions

In this paper an unrelated parallel machine scheduling problem that considers several important characteristics arising in real world manufacturing environments was approached. In particular, we addressed the job splitting property that can contribute significantly to the reduction of the production lead time in parallel machine environments, as for example in the textile industry or in the electronic industry, thus contributing to the reduction of tardy jobs. Nonetheless, this property received little attention by the scheduling research community and only in the last recent years a few papers about parallel machine scheduling considering this property were published.

Another important feature of the practical scheduling problems, particularly in the recent years due to the proliferation of just-in-time production practices and lean manufacturing, is the objective to guarantee the processing of the customers' orders as close as possible to their due dates, thus preventing unnecessary work in progress (WIP), inventory costs and customer dissatisfaction caused by late deliveries. The objective function considered in this work (in both approaches proposed) minimizes both WIP and late deliveries, being an important contribution of this work, as this type of objective was seldom studied in the literature so far.

Two models were proposed: an integer programming compact model and a decomposition model. The first one was solved directly through a state-of-the-art commercial general purpose mixed integer programming solver and the second one was approached by combining column generation and the same general purpose mixed integer programming solver through SearchCol framework.

Computational results on instances up to 10 machines and 100 jobs show that the compact model approach is, in general, much more efficient, although for some of the largest and most congested instances no feasible solutions were obtained unlike the SearchCol algorithm used for the decomposition model.

Acknowledgements This work is financed by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projects "SearchCol: Metaheuristic search by column generation" (PTDC/EIAEIA/100645/2008) and PEst-OE/EEI/UI0319/2014.

References

1. Allahverdi, A., Gupta, J.N., Aldowaisan, T.: A review of scheduling research involving setup considerations. Omega **27**(2), 219–239 (1999)
2. Allahverdi, A., Ng, C., Cheng, T., Kovalyov, M.: A survey of scheduling problems with setup times or costs. Eur. J. Oper. Res. **187**(3), 985–1032 (2008)
3. Alvelos, F., Sousa, A., Santos, D.: Combining column generation and metaheuristics. In: Talbi, E.G. (ed.) Hybrid Metaheuristics. Studies in Computational Intelligence, vol. 434, pp. 285–334. Springer, Berlin/Heidelberg (2013)
4. Blum, C., Puchinger, J., Raidl, G.R., Roli, A.: Hybrid metaheuristics in combinatorial optimization: a survey. Appl. Soft Comput. **11**(6), 4135–4151 (2011)

5. Chen, J.F.: Scheduling on unrelated parallel machines with sequence-and machine-dependent setup times and due-date constraints. Int. J. Adv. Manuf. Technol. **44**(11), 1204–1212 (2009)
6. Chen, J.F., Wu, T.H.: Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints. Omega **34**(1), 81–89 (2006)
7. Dantzig, G., Wolfe, P.: Decomposition principle for linear programs. Oper. Res. **8**, 101–111 (1960)
8. Desaulniers, G., Desrosiers, J., Solomon, M.: Column Generation, vol. 5. Springer, New York (2005)
9. Desrosiers, J., Lübbecke, M.: A primer in column generation. In: Column Generation, pp. 1–32. Springer, New York (2005)
10. Fanjul-Peyro, L., Ruiz, R.: Scheduling unrelated parallel machines with optional machines and jobs selection. Comput. Oper. Res. **39**, 1745–1753 (2012)
11. Florêncio, L.: A searchcol algorithm for the unrelated parallel machine scheduling problem with job splitting. Master's thesis, Universidade do Minho (2013)
12. ILOG: IBM ILOG CPLEX Optimization Studio V12.2 (2010)
13. Kim, D.W., Kim, K.H., Jang, W., Frank Chen, F.: Unrelated parallel machine scheduling with setup times using simulated annealing. Robot. Comput.-Integr. Manuf. **18**(3), 223–231 (2002)
14. Kim, Y., Shim, S., Kim, S., Choi, Y., Yoon, H.: Parallel machine scheduling considering a job-splitting property. Int. J. Prod. Res. **42**(21), 4531–4546 (2004)
15. Lee, J.H., Yu, J.M., Lee, D.H.: A tabu search algorithm for unrelated parallel machine scheduling with sequence- and machine-dependent setups: minimizing total tardiness. Int. J. Adv. Manuf. Technol. **69**(9–12), 2081–2089 (2013)
16. Liaw, C.F., Lin, Y.K., Cheng, C.Y., Chen, M.: Scheduling unrelated parallel machines to minimize total weighted tardiness. Comput. Oper. Res. **30**(12), 1777–1789 (2003)
17. Lin, Y., Pfund, M., Fowler, J.: Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. Comput. Oper. Res. **38**, 901–9016 (2011)
18. Logendran, R., Subur, F.: Unrelated parallel machine scheduling with job splitting. IIE Trans. **36**(4), 359–372 (2004)
19. Logendran, R., McDonell, B., Smucker, B.: Scheduling unrelated parallel machines with sequence-dependent setups. Comput. Oper. Res. **34**(11), 3420–3438 (2007)
20. Lopes, M.P., Carvalho, J.d.: A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times. Eur. J. Oper. Res. **176**(3), 1508–1527 (2007)
21. Nait, T.D., Yalaoui, F., Chu, C., Amodeo, L.: A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. Int. J. Prod. Econ. **99**(1), 63–73 (2006)
22. Park, T., Lee, T., Kim, C.O.: Due-date scheduling on parallel machines with job splitting and sequence-dependent major/minor setup times. Int. J. Adv. Manuf. Technol. **59**(1), 325–333 (2012)
23. Pfund, M., Fowler, J.W., Gupta, J.N.: A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems. J. Chin. Inst. Ind. Eng. **21**(3), 230–241 (2004)
24. Pinedo, M.L.: Scheduling: Theory, Algorithms, and Systems, 2nd edn. Springer, New York (2002)
25. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: a review. Eur. J. Oper. Res. **120**(2), 228–249 (2000)
26. Rocha, P., Ravetti, M., Mateus, G., Pardalos, P.: Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. Comput. Oper. Res. **35**(4), 1250–1264 (2008)
27. Rodriguez, F.J., Lozano, M., Blum, C., García-Martínez, C.: Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. Comput. Oper. Res. **40**, 1829–1841 (2013)
28. Sarıçiçek, İ., Çelik, C.: Two meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness. Appl. Math. Model. **35**(8), 4117–4126 (2011)

29. Shim, S.O., Kim, Y.D.: Minimizing total tardiness in an unrelated parallel-machine scheduling problem. J. Oper. Res. Soc. **58**(3), 346–354 (2006)
30. Shim, S., Kim, Y.: A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property. Comput. Oper. Res. **35**(3), 863–875 (2008)
31. Talbi, E.G.: Hybrid Metaheuristics. Studies in Computational Intelligence, vol. 434. Springer, Berlin/Heidelberg (2013)
32. Unlu, Y., Mason, S.: Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. Comput. Ind. Eng. **58**(4), 785–800 (2010)
33. Vallada, E., Ruiz, R.: A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. Eur. J. Oper. Res. **211**(3), 612–622 (2011)
34. Wang, W.L., Wang, H.Y., Zhao, Y.W., Zhang, L.P., Xu, X.L.: Parallel machine scheduling with splitting jobs by a hybrid differential evolution algorithm. Comput. Oper. Res. **40**(5), 1196–1206 (2013)
35. Xing, W., Zhang, J.: Parallel machine scheduling with splitting jobs. Discret. Appl. Math. **103**(1), 259–269 (2000)
36. Yalaoui, F., Chu, C.: An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times. IIE Trans. **35**(2), 183–190 (2003)
37. Yang, W.H.: Survey of scheduling research involving setup times. Int. J. Syst. Sci. **30**(2), 143–155 (1999)
38. Zhu, Z., Heady, R.: Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. Comput. Ind. Eng. **38**(2), 297–305 (2000)

Testing Regularity on Linear Semidefinite Optimization Problems

Eloísa Macedo

Abstract This paper presents a study of regularity of Semidefinite Programming (SDP) problems. Current methods for SDP rely on assumptions of regularity such as constraint qualifications (CQ) and well-posedness. In the absence of regularity, the characterization of optimality may fail and the convergence of algorithms is not guaranteed. Therefore, it is important to have procedures that verify the regularity of a given problem before applying any (standard) SDP solver. We suggest a simple numerical procedure to test within a desired accuracy if a given SDP problem is regular in terms of the fulfilment of the Slater CQ. Our procedure is based on the recently proposed DIIS algorithm that determines the immobile index subspace for SDP. We use this algorithm in a framework of an interactive decision support system. Numerical results using SDP problems from the literature and instances from the SDPLIB suite are presented, and a comparative analysis with other results on regularity available in the literature is made.

1 Introduction

Linear Semidefinite Programming (SDP) deals with problems of minimization/maximization of a linear objective function subject to constraints in the form of linear matrix inequalities. SDP can be considered as a generalization of Linear Programming (LP), where matrices are used instead of vectors. Recently, special attention has been devoted to SDP due to many applications in engineering, control theory, statistics, financial models and combinatorial optimization [33].

Most of the well-known and efficient methods for SDP, the duality theory and SDP optimality conditions rely on assumptions of regularity [6, 15, 33]. The lack of regularity significantly affects the characterization of optimality of a solution. With respect to algorithms, the regularity of a problem is a condition that guarantees their stability and efficiency. In the absence of regularity, a problem may be poorly behaved and the resulting solution may be corrupted [6].

E. Macedo (✉)
University of Aveiro, Campus Universitário de Santiago, Aveiro, Portugal
e-mail: macedo@ua.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_13

In the literature, different concepts are being associated to the notion of regularity. An optimization problem is commonly considered to be regular if certain constraint qualification (CQ) is satisfied, and nonregular, otherwise [12]. Regularity conditions play also an important role in deriving duality relations, sensitivity/stability analysis and convergence of computational methods [14]. Many optimality conditions, such as the classical Karush-Kuhn-Tucker (KKT) optimality conditions, are formulated under the fulfilment of certain CQs. One of such conditions is the Slater CQ (see [4, 15]) that consists in the nonemptiness of the interior of the feasible set. According to [4], there are many instances of SDP problems for which Slater CQ fails to hold, leading to numerical difficulties when standard SDP solvers are applied. Therefore, it is important to know in advance if a given problem satisfies the Slater CQ, in order to avoid poor behaviour of numerical methods. To our knowledge, there is no simple numerical procedure to test the Slater CQ. This is one of the purposes of the present work.

Another kind of regularity notion in Optimization and Numerical Methods is known as well-posedness of a problem. In general, one can define well-posedness of a problem in the sense of Hadamard or in the sense of Tikhonov. According to [5, 18], an optimization problem is well-posed in the sense of Hadamard if it has a unique solution that is stable, i.e., depends continuously on data, meaning that small perturbations on data give rise to small variations on the solution. According to [5, 18, 29], a problem is well-posed in the sense of Tikhonov if it has a unique solution and every minimizing sequence for the optimization problem converges to that solution. The theoretical study of well-posedness of certain classes of optimization problems is a rather difficult issue. A problem that is not well-posed is called ill-posed. This kind of problems are quite common in applications and, according to [13], the ill-posedness may occur, for instance, due to the lack of precise mathematical formulations.

In [8, 13], a practical characterization of well-posedness of conic and in particular, SDP problems is proposed and it is based on a so called condition number defined by Renegar in [27]. It is showed that the Renegar's condition number is infinite if and only if the problem is ill-posed. The Renegar's condition number is defined as a scale-invariant reciprocal of a problem instance to be infeasible. Therefore, a SDP problem is considered to be well-posed if its Renegar condition number is finite, and ill-posed, otherwise. In [13], the calculus of this condition number is connected with upper bounds of optimal values of SDP problems. The approach proposed in [13] for characterization of regularity is constructive and based on obtaining rigorous bounds and also error bounds for the optimal values, by properly postprocessing the output of a SDP solver.

There exist some studies dedicated to interrelation between regularity and well-posedness of optimization problems. In [18], different notions of well-posedness of general convex problems are studied and compared and it is also shown that under the Slater CQ, Hadamard's well-posedness is equivalent to that of Tikhonov. In particular, it is mentioned that uniqueness of the solution of a finite convex minimizing problem is enough to guarantee that the problem is Tikhonov well-posed, and that this is no longer valid for the infinite case. Other definitions of

well-posedness are considered in [18] as well, namely, the Levitin-Polyak well-posedness and strong well-posedness. It is proved that well-posedness in the sense of Hadamard implies either Tykhonov, Levitin-Polyak and strong well-posedness. In [11, 13, 32] the relationship between well-posedness and regularity in the sense of the fulfilment of the Slater CQ is considered for SDP problems.

In [19], it is suggested the DIIS algorithm (stands for Determination of the Immobile Index Subspace) to find a basis of the subspace of immobile indices, which is showed to be an important characteristic of the feasible set permitting to develop new CQ-free optimality conditions. Moreover, in [19], it is proved that if the subspace of immobile indices is null, then the SDP problem satisfies the Slater CQ. We have numerically implemented the DIIS algorithm and tested it using SDP problems from the literature and from SDPLIB. Since each iteration of the DIIS algorithm is based on the solution of a quadratic system of equations, we suggest and discuss two different approaches to address its solution. The obtained numerical results on regularity in terms of the fulfilment of the Slater CQ were compared with other regularity tests in terms of well-posedness described in [8] and [13].

The paper is organized as follows. The basic definitions and the study of different notions of regularity of linear SDP problems are presented in Sect. 2. The description of the algorithm to test the Slater CQ for these problems is carried out in Sect. 3. In Sect. 4, new approaches that can be used to test the Slater CQ are suggested and a numerical procedure to test regularity is presented in Sect. 5. The numerical results as well as the conclusions of the experiments are presented in Sect. 6.

2 Regularity in SDP

This section begins with basic definitions of SDP and is devoted to the study of different notions of regularity in SDP.

2.1 Constraint Qualifications for Linear Semidefinite Programming Problems

Given $s \in \mathbb{N}$, denote by $S(s)$ the space of $s \times s$ real symmetric matrices endowed with the trace inner product defined by

$$tr(AB) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ji}$$

for $A, B \in S(s)$. A matrix $A \in S(s)$ is positive semidefinite ($A \succeq 0$) if $x^T Ax \geq 0, \forall x \in \mathbb{R}^s$ and a matrix $A \in S(s)$ is negative semidefinite ($A \preceq 0$) if $-A \succeq 0$. Let $P(s) \subset S(s)$ be the cone of positive semidefinite symmetric $s \times s$ matrices.

Consider the linear SDP problem

$$\min c^T x, \quad \text{s. t. } \mathscr{A}(x) \preceq 0, \tag{1}$$

where $x \in \mathbb{R}^n$, and the matrix-valued function $\mathscr{A}(x)$ is defined as $\mathscr{A}(x) := \sum_{i=1}^{n} A_i x_i + A_0, A_i \in S(s), i = 0, 1, \ldots, n$.

The SDP problem (1) is a convex problem and its (convex) feasible set is given by

$$\mathscr{X} = \{x \in \mathbb{R}^n : \mathscr{A}(x) \preceq 0\}.$$

The Lagrangian dual of problem (1) is given by

$$\max \ tr(A_0 Z), \quad \text{s. t. } -tr(A_i Z) = c_i, \forall i = 1, \ldots, n, \quad Z \succeq 0, \tag{2}$$

where $Z \in P(s)$. The feasible set of problem (2) is

$$Z = \{Z \in P(s) : -tr(A_i Z) = c_i, i = 1, \ldots, n\}.$$

We will refer to problem (1) as the primal problem and to problem (2) as the dual one. Notice here that some authors consider that the primal SDP problem has the form (2), and the dual problem has the form (1). This is not an issue, since it is possible to transform a SDP problem in the form (1) into the form (2), and vice-versa (see [33]).

The duality results in SDP are more subtle than in Linear Programming (LP). Nevertheless, the following property of LP problems still holds for SDP, inducing a lower bound on the value of the primal problem:

Theorem 1 (Weak Duality) *Given a pair of primal and dual feasible solutions $x \in \mathscr{X}, Z \in \mathscr{Z}$ of SDP problems (1) and (2), the inequality $c^T x \geq tr(A_0 Z)$ always holds.*

Proof Considering the formulation of the problems (1) and (2), we have that

$$c^T x - tr(ZA_0) = \sum_{i=1}^{n} (-tr(ZA_i)) x_i - tr(ZA_0) = tr\left(\left(-A_0 - \sum_{i=1}^{n} A_i x_i\right) Z\right).$$

Since $-A_0 - \sum_{i=1}^{n} A_i x_i \succeq 0$ and $Z \succeq 0$, then $tr\left(\left(-A_0 - \sum_{i=1}^{n} A_i x_i\right) Z\right) \geq 0$, and thus, we obtain the inequality $c^T x \geq tr(A_0 Z)$. □

Definition 1 Denote by p^* the optimal value of the objective function of the primal SDP problem (1) and by d^* the optimal value of the objective function of the dual problem (2). The difference $p^* - d^*$ is called duality gap.

Unlike LP, a nonzero duality gap can occur in SDP and to guarantee strong duality some additional assumptions, such as the fulfilment of some constraint qualification (CQ), have to be made [1, 32]. Constraint qualifications are conditions imposed on the constraints of an optimization problem that should be satisfied in order to apply the Karush-Kuhn-Tucker optimality conditions. There exist several CQs, such as the Mangasarian-Fromovitz CQ, the Robinson CQ, and the Slater CQ. According to [4], the Robinson CQ is equivalent to the Mangasarian-Fromovitz CQ in the case of conventional nonlinear programming, which includes convex linear SDP. In [33], it is proved that in the particular case of convex SDP, the Slater CQ is equivalent to the Robinson CQ. Since the Slater CQ implies the Robinson CQ, then it also implies the Mangasarian-Fromovitz CQ, meaning that the Slater CQ is a stronger CQ.

One of the most widely known CQ in finite and infinite optimization is the Slater CQ. In SDP, many authors assume in their studies that this condition holds (see [4, 11, 15, 32]).

Definition 2 The constraints of the SDP problem (1) satisfy the Slater CQ if the feasible set $\mathscr{X}$ has a nonempty interior, i.e., $\exists\, \bar{x} \in \mathbb{R}^n : \mathscr{A}(\bar{x}) \prec 0$.

Here, $A \prec 0$ ($A \succ 0$) denotes that matrix $A \in S(s)$ is negative (positive) definite.

The analogous definition can be introduced for the dual SDP problem.

Definition 3 The constraints of the dual SDP problem (2) satisfy the Slater CQ if there exists a matrix $Z \in P(s)$, such that $-tr\,(A_i Z) = c_i$, $\forall i = 1, \ldots, n$ and $Z \succ 0$.

The Slater CQ is sometimes called strict feasibility [32] or Slater regularity condition [15].

If problem (1) satisfies the Slater CQ, then the following property is valid ([4, 32]):

Theorem 2 (Strong Duality) *Under the Slater CQ for the SDP problem* (1)*, if the primal optimal value is finite, then the duality gap vanishes and the (dual) optimal value of problem* (2) *is attained.*

The first order necessary and sufficient optimality conditions for linear SDP can be formulated, under the fulfilment of the Slater CQ, in the form of the following theorem from [2]:

Theorem 3 *If problem* (1) *satisfies the Slater CQ, then* $x^* \in \mathscr{X}$ *is an optimal solution if and only if there exists a matrix* $Z^* \in P(s)$ *such that*

$$tr\left(Z^* A_i\right) + c_i = 0,\ i = 1, \ldots, n \ \text{ and } \ tr\left(Z^* \mathscr{A}(x^*)\right) = 0. \tag{3}$$

In SDP, if the Slater CQ does not hold, some pathologies may occur: the dual (or primal) optimal value may not be attained, one of the problems may be feasible

and bounded, while the other is infeasible, or a nonzero duality gap can exist and conditions (3) (also called complementary conditions) may fail [4, 19, 32]. Hence, the complete characterization of optimality of feasible solutions may be compromised.

2.2 Well-Posedness of Linear Semidefinite Programming Problems

In [8] and [13], two constructive approaches to classify SDP problems in terms of well-posedness are proposed, both based on the concept of well-posedness in the sense defined by Renegar in [27].

The Renegar condition number of a problem's instance is defined as a scale-invariant reciprocal of the distance to infeasibility (the smallest data perturbation that results in either primal or dual infeasibility). According to [8], the Renegar's condition number for problem (1) is defined by

$$C(d) := \frac{\|d\|}{\min\{\rho_P(d), \rho_D(d)\}},$$

where $\|d\| := \max\{\|\mathscr{A}\|, \|c\|, \|A_0\|\}$ is the norm on the data space d, $\rho_P(d)$ and $\rho_D(d)$ are the distance to primal and dual infeasibility, respectively. The distance to primal infeasibility is defined as

$$\rho_P(d) := \inf\{\|\triangle d\| : \mathscr{X}_{\triangle d} = \emptyset\},$$

where $\triangle d$ is the perturbation on data and $\mathscr{X}_{\triangle d}$ is the feasible region of the perturbed SDP problem and the distance to dual infeasibility $\rho_D(d)$ is defined in a similar way, but considering the dual problem. The SDP problem is considered to be ill-posed (in the sense of Renegar) if $C = \infty$.

The numerical approach described in [8] is based on the estimation of lower and upper bounds of the Renegar's condition number C of a SDP problem. The distance to primal infeasibility is obtained by solving several auxiliary SDP problems of compatible size to the original SDP problem and the distance to dual infeasibility is found by solving one single SDP auxiliary problem. According to Proposition 3 in [8], the estimation of the norm of data can be done with the help of its upper and lower bounds using straightforward matrix norms and maximum eigenvalue computations. Notice that it is necessary to choose adequate norms for computing the distances to primal and dual infeasibility, as well as a SDP solver.

In [13], the characterization of well-posedness of SDP problems is based on the calculus of rigorous lower and upper bounds of their optimal values. It is shown that if the rigorous upper bound $\bar{p}^*$ of the primal objective function is infinite, then the Renegar condition number C is also infinite and hence, the SDP problem is ill-posed.

An algorithm for computing the upper bound $\bar{p}^*$ (Algorithm 4.1 from [13]) is described in [13]. On its iterations, some auxiliary perturbed "midpoint" SDP problems are solved using a SDP solver and special interval matrices are constructed on the basis of their solutions. The constructed interval matrices must contain a primal feasible solution of the perturbed "midpoint" problem and satisfy the conditions (4.1) and (4.2) of Theorem 4.1 from [13]. If such interval matrix can be computed, then the optimal value of the primal objective function of the SDP problem is bounded from above by $\bar{p}^*$, which is the primal objective function value considering the obtained interval matrix. Besides requiring a SDP solver for computing the approximate solutions of the perturbed problems, the approach proposed in [13] also needs verified solvers for interval linear systems and eigenvalue problems.

2.3 Relationship Between Regularity Notions in SDP

Nevertheless the above considered definitions of regularity of SDP problems are different, there exist a deep connection between them. According to [32], the lack of regularity in terms of the fulfilment of the Slater CQ implies ill-posedness of the problem.

The following lemma can be easily proved.

Lemma 1 *If a linear SDP problem in the form* (1) *does not satisfy the Slater CQ, then it is ill-posed.*

Proof Indeed, since the Slater CQ is not satisfied, all the feasible solutions of the SDP problem lie on the boundary of the feasible set. Hence, any small data perturbations may lead to the loss of feasibility, meaning that the problem is ill-posed.

□

Notice that the reciprocal of Lemma 1 is not true. The following example shows that there exist problems that are ill-posed, but do satisfy the Slater CQ.

Example 1 Consider the primal SDP problem

$$\begin{array}{l} \min\ x_1 - x_2 - x_3 \\ \text{s.t.} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} x_3 + \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \preceq 0. \end{array} \tag{4}$$

The dual problem to (4) has the form

$$\begin{array}{l}\max\ y_1 + y_2 \\ \text{s.t.} \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_1 + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_2 + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} y_3 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} y_4 + \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \preceq 0.\end{array}$$

The constraints of the primal problem (4) satisfy the Slater CQ, since there exists a strictly feasible solution: e.g., $x_1 = 1, x_2 = 2$ and $x_3 = 1$. However, problem (4) is ill-posed, since its Renegar condition number is infinite. Indeed, it is easy to see that the dual of the problem (4) is infeasible, i.e., there is no possible feasible solution satisfying the constraints, and hence, the distance to dual infeasibility is zero.

3 The Algorithm of Determination of the Immobile Index Subspace

In this section, we introduce the notions of immobile indices and immobile index subspace and describe the algorithm of their determination that can be used to verify if a given SDP problem satisfy the Slater CQ.

3.1 Subspace of Immobile Indices

Given the linear SDP problem (1) it is easy to see that it is equivalent to the following convex Semi-Infinite Programming (SIP) problem:

$$\min c^T x, \quad \text{s. t. } l^T \mathscr{A}(x) l \leq 0, \forall l \in L := \{l \in \mathbb{R}^s : \|l\| = 1\}, \tag{5}$$

where $L \subset \mathbb{R}^s$ is a s-dimensional index set. The (convex) feasible set of problem (5) is $\left\{x \in \mathbb{R}^n : l^T \mathscr{A}(x) l \leq 0, \forall l \in L\right\}$. Notice that it coincides with the feasible set of problem (1), $\mathscr{X}$. Therefore, problems (1) and (5) are equivalent.

Since our approach to test the Slater CQ on SDP programs involves an equivalent formulation as a SIP problem, it is worth introduce the following definition.

Definition 4 The SIP problem (5) satisfies the Slater CQ if there exists a feasible point $\bar{x} \in \mathbb{R}^n$ such that the inequalities $l^T \mathscr{A}(\bar{x}) l < 0$ hold, for all indices $l \in L$.

The approach that we propose here to test the regularity in SDP is based on the notions of immobile indices for SIP and subspace of immobile indices for SDP. These notions were suggested in [19] and showed to be important to obtain new optimality conditions for linear SDP.

Definition 5 Given a convex SIP problem (5), an index $l^* \in L$ is called immobile if $l^{*T}\mathscr{A}(x)l^* = 0, \forall x \in \mathscr{X}$.

The set of immobile indices of the SIP problem (5) is given by

$$L^* = \left\{l \in L : l^T \mathscr{A}(x)l = 0, \forall x \in \mathscr{X}\right\}.$$

It is proved in [19] that, given the pair of equivalent problems (1) and (5), the set of immobile indices L^* of problem (5) can be presented in the form $L^* = L \cap \mathscr{M}$, where $\mathscr{M}$ is a subspace of $\mathbb{R}^s$ defined by

$$\mathscr{M} := \left\{l \in \mathbb{R}^s : l^T \mathscr{A}(x)l = 0, \forall x \in \mathscr{X}\right\} = \left\{l \in \mathbb{R}^s : \mathscr{A}(x)l = 0, \forall x \in \mathscr{X}\right\}. \tag{6}$$

The subspace $\mathscr{M}$ is called *subspace of immobile indices* of the SDP problem (1).

It is easy to see that the equivalent pair of problems (1) and (5) satisfy or not the Slater CQ, simultaneously. The following propositions are proved in [19].

Proposition 1 *The convex SIP problem* (5) *satisfies the Slater CQ if and only if the set* L^* *is empty.*

Proposition 2 *The SDP problem* (1) *satisfies the Slater CQ if and only if the set of immobile indices in the corresponding SIP problem* (5) *is empty.*

Notice that the set of immobile indices L^* for the SIP problem (5) is empty if and only if the subspace of immobile indices $\mathscr{M}$ for the SDP problem (1) is null.

From Proposition 1, it follows that problem (1) is regular if and only if $\mathscr{M}$ is null, i.e., $\mathscr{M} = \{0\}$. The dimension of the subspace $\mathscr{M}$, denoted by s^*, can be considered as an irregularity degree of the SDP problem (1). Moreover,

- if $s^* = 0$, then the problem is regular, i.e., the Slater CQ holds;
- if $s^* = 1$, then the problem is nonregular, with minimal irregularity degree;
- if $s^* = s$, then the problem is nonregular, with maximal irregularity degree.

In [19], it is shown that the subspace of immobile indices plays an important role in characterization of optimality of SDP problems and a new CQ-free optimality criterion is formulated, based on the explicit determination of the subspace of immobile indices $\mathscr{M}$. The constructive algorithm (the DIIS algorithm) that finds a basis of the subspace $\mathscr{M}$ is described and justified in [19]. We will use the DIIS algorithm to verify if the Slater CQ holds for a given SDP problem, permitting to conclude about its regularity.

3.2 The DIIS Algorithm

Consider a linear SDP problem (1) with $\mathscr{A}(x)$ defined by $n + 1$ symmetric $s \times s$ matrices A_i, $i = 0, 1, \ldots, n$ and suppose that its feasible set is nonempty. The DIIS algorithm proposed in [19] constructs a basis $M = (m_i, i = 1, \ldots, s^*)$ of the

subspace of immobile indices $\mathscr{M}$. At the k-th iteration, I^k denotes some auxiliary set of indices and M^k denotes an auxiliary set of vectors. Suppose that $s > 1$, with $s \in \mathbb{N}$.

DIIS algorithm

input: $s \times s$ symmetric matrices $A_j, j = 0, 1, \ldots, n$.
Set $k := 1$, $I^1 := \emptyset$, $M^1 := \emptyset$.
repeat given k, I^k, M^k:
set $p_k := s - \left|I^k\right|$
solve the quadratic system:

$$\begin{cases} \sum\limits_{i=1}^{p_k} l_i^T A_j l_i + \sum\limits_{i \in I^k} \gamma_i^T A_j m_i = 0, \; j = 0, 1, \ldots, n, \\ \sum\limits_{i=1}^{p_k} \|l_i\|^2 = 1, \\ l_i^T m_j = 0 \; , \; j \in I^k, \; i = 1, \ldots, \; p_k, \end{cases} \tag{7}$$

where $l_i \in \mathbb{R}^s, i = 1, \ldots, p_k$ and $\gamma_i \in \mathbb{R}^s, i \in I^k$
if system (7) does not have a solution, **then** stop and return the current k, I^k and M^k.
else given the solution $\left\{l_i \in \mathbb{R}^s, i = 1, \ldots, p_k, \gamma_i \in \mathbb{R}^s, i \in I^k\right\}$ of (7):
construct the maximal subset of linearly independent vectors

$$\{m_1, \ldots, m_{s_k}\} \subset \left\{l_1, \ldots, l_{p_k}\right\}$$

update:

$$\triangle I^k := \left\{\left|I^k\right| + 1, \ldots, \left|I^k\right| + s_k\right\}$$
$$M^{k+1} := M^k \cup \left\{m_j, \; j \in \triangle I^k\right\}$$
$$I^{k+1} := I^k \cup \triangle I^k.$$

do $k := k + 1$

In [19], it is proved that the DIIS algorithm returns the set of immobile indices I^k and the basis M^k of the subspace of immobile indices $\mathscr{M}$.

Considering the results presented in the previous Sect. 3.1 and the properties of the DIIS algorithm in [19], we can conclude that:

- if the Slater CQ is satisfied, then the DIIS algorithm stops at the first iteration with $k = 1$, $\mathscr{M} = \{0\}$ and $s^* = 0$;
- if the Slater CQ is violated, then the algorithm returns a basis $M = M^k$, such that $rank(M) = s^* > 0$.

Notice that the main procedure on each iteration of the DIIS algorithm is to solve the quadratic system of Eqs. (7). At each iteration k, this system has $p_k + |I^k|$ vector variables (and since each vector variable has s components, one has $s(p_k + |I^k|)$ scalar variables) and $n + 2 + p_k \times |I^k|$ equations. Notice also that one iteration is enough to verify the fulfilment of the Slater CQ on SDP problems and in this case, one has s vector variables and $n + 2$ equations.

Considering the description of the DIIS algorithm, we can conclude that technically, testing of regularity in terms of the Slater CQ on SDP problems using the DIIS algorithm from [19] is more simple than testing their well-posedness using the methods from [8] and [13], since these last methods present a more complex framework. To classify SDP problems using the methods from [8] and [13], several auxiliary SDP problems must be solved using standard SDP solvers, and the method proposed in [13] also involves interval arithmetic and the solution of eigenvalue problems.

4 New Approaches to Test Regularity of Linear Semidefinite Programming Problems

The numerical procedure we suggest to test the regularity on SDP problems in terms of the fulfilment of the Slater CQ uses an adaptation of the DIIS algorithm [19] and is based on a cross-check of two numerical approaches for solving the quadratic system of equations.

Taking into account the description of the DIIS algorithm, it is enough to compute only one iteration of the algorithm to verify the Slater CQ on SDP problems.

At the first iteration, the basis of the subspace $\mathscr{M}$ is empty, $p_1 = s$ and we solve the following quadratic system of equations

$$\begin{cases} \sum_{i=1}^{s} l_i^T A_j l_i = 0, \ j = 0, 1, \ldots, n, \\ \sum_{i=1}^{s} \|l_i\|_2^2 = 1, \end{cases} \tag{8}$$

where $l_i \in \mathbb{R}^s$, $i = 1, \ldots, s$, are the variables, $A_j \in S(s)$, $j = 0, 1, \ldots, n$ and n is the dimension of the variable space of the SDP problem (1).

If system (8) is consistent, then the dimension s^* of the subspace of immobile indices $\mathscr{M}$ is nonzero, otherwise, $\mathscr{M} = \{0\}$ and $s^* = 0$. In the first case, one can conclude that the Slater CQ is violated, while in the latter, one can conclude that the Slater CQ is satisfied.

The important question here is how to ensure that system (8) is not consistent, i.e., does not admit any solution. Adequate numerical procedures should be used to verify this issue.

Exact solving of the system (8), i.e., obtaining its exact solution, can be rather difficult. The numerical solution procedures for nonlinear systems are iterative and rely on approximate solutions. In our case, we are interested in obtaining accurate solutions for the system (8), or guaranteeing that the system is not consistent, within a desired degree of certainty. We will present two numerical approaches that can be used to ensure that system (8) is consistent or not within a desirable tolerance, and then be able to conclude about the regularity of a SDP problem in the form (1) in terms of the fulfilment of Slater's CQ.

Numerical Approach I

For the sake of clarity, let us rewrite system (8) in the following componentwise form

$$\begin{cases} F_j(\ell) \quad = \sum_{i=1}^{s} l_i^T A_j l_i = 0, \; j = 0, 1, \ldots, n, \\ F_{n+1}(\ell) = \sum_{i=1}^{s} \|l_i\|_2^2 - 1 \; = \; 0, \end{cases} \tag{9}$$

where vector ℓ is defined by a concatenation of the vectors $l_1, l_2, \ldots, l_s \in \mathbb{R}^s$, each one with s scalar variables as follows:

$$\ell = \begin{bmatrix} l_1 \; l_2 \; \ldots \; l_s \end{bmatrix}^T = \begin{bmatrix} \ell_1 \; \ell_2 \; \ldots \; \ell_m \end{bmatrix}^T \in \mathbb{R}^m,$$

where $\ell_i \in \mathbb{R}$, $i = 1, \ldots, m$, and $m = s^2$.

System (9) has $n + 2$ nonlinear equations and m unknowns. In general, we have $m \geq n + 2$ and thus, it is an underdetermined system. Obviously, the functions F_j, $j = 0, 1, \ldots, n+1$ are quadratic real valued functions defined in $\mathbb{R}^m$, hence they are continuous and smooth. Although for every $j = 0, 1, \ldots, n$, the function F_j is convex if and only if the symmetric matrix A_j is positive semidefinite, which may not hold. Evidently, the function F_{n+1} is convex.

Any solution of system (9) is a minimizer of the function $\sum_{j=0}^{n+1} F_j^2(\ell)$ and thus, we can formulate the following unconstrained nonlinear least-squares problem:

$$\min_{\ell \in \mathbb{R}^m} G(\ell) = \sum_{j=0}^{n+1} F_j^2(\ell). \tag{10}$$

Notice that $G(\ell) \geq 0, \; \forall \ell \in \mathbb{R}^m$.

Considering problem (10) and denoting by ℓ^* its solution, the following two situations can occur:

- if $G(\ell^*) = 0$, then ℓ^* is a solution of system (9) and, consequently, system (8) is consistent. Therefore, the SDP problem (1) does not satisfy the Slater CQ;
- if $G(\ell^*) > 0$, then system (9) is not consistent, as well as system (8) and one can conclude that Slater's CQ holds for the SDP problem (1).

Notice also that problem (10) is a global optimization problem and may have multiple local minima. Therefore, its solution may be not unique. There exist several algorithms to solve the nonlinear least-squares problem (10), namely, the Levenberg-Marquardt Algorithm, the Gauss-Newton Algorithm and the Trust-Region-Reflective Algorithm [22, 23]. These algorithms are already implemented in the MATLAB function `lsqnonlin`.

Since problem (10) may have multiple local minima, the algorithms may not reach the global minimum. Running the algorithm with several starting points may increase the degree of certainty that the system (9) is not consistent. Moreover,

the cross-check of results obtained by different algorithms may also increase that certainty.

Numerical Approach II

Another numerical approach to solve system (8) is based on solving a nonlinear programming problem with equality constraints.

Consider the following problem:

$$\begin{aligned} \min_{y\in\mathbb{R}^{n+2}, l_i\in\mathbb{R}^s} \quad & H(y) = \|y\|_2^2 \\ \text{s.t.} \quad & \sum_{i=1}^{s} l_i^T A_j l_i + y_{j+1} = 0,\ j = 0, 1, \ldots, n, \\ & \sum_{i=1}^{s} \|l_i\|_2^2 + y_{n+2} - 1 = 0, \end{aligned} \tag{11}$$

where $y = \left[y_1\ y_2 \cdots y_{n+2} \right]^T$.

Let $y^* \in \mathbb{R}^{n+2}$ be a solution of problem (11). The following two situations can occur:

- if $H(y^*) > 0$, then system (8) is not consistent and the SDP problem (1) satisfies the Slater CQ;
- if $H(y^*) = 0$, then system (8) is consistent and we can conclude that the Slater CQ is violated.

The problem (11) can be easily solved using interior point methods, which perform very well in practice. These methods are based on a barrier function and keep all the iterates in the interior of the feasible set and are specially useful for problems with sparse data or problems that have a particular structure. The routine `fmincon` from MATLAB uses several algorithms to solve constrained minimization problems. Besides an implementation of an Interior Point Algorithm, the routine also may use a Trust-Region-Reflective Algorithm, Active-Set Algorithm and a Sequential Quadratic Programming Algorithm.

5 Numerical Procedure to Test the Fulfilment of the Slater CQ

To test regularity in terms of the fulfilment of the Slater CQ, a routine in MATLAB language has been created. This routine is the implementation of the DIIS algorithm. On its iterations, the existence of solution of the quadratic system (8) is verified using both approaches, I and II. The numerical procedure allows the user to specify which of the two approaches to use. The implementation of both numerical approaches, Approach I and Approach II, is included to permit a cross-check of results. The cross-check of different numerical approaches is important to increase the reliability on the solvers approximate solution.

In order to increase the reliability on the results obtained using the suggested numerical approaches, the following techniques were used in the initial testing phase:

- each solver was run 10 times with different random starting points for each problem;
- each solver was restarted using the last approximation computed earlier;
- different tolerances values were tried;
- both numerical and analytical Jacobians were used.

Experiments showed that the most powerful technique is running the solver several times with different starting points. By default, our program uses a randomly chosen starting point.

The program stops when one of the following stopping criteria is satisfied:

- $||F(\ell^{(i)}) - F(\ell^{(i+1)})||_\infty < \texttt{TolFun}$, where `TolFun` is a tolerance on the function value;
- $||\ell^{(i)} - \ell^{(i+1)}||_\infty < \texttt{TolX}$, where `TolX` is a tolerance on the argument variable value;
- $||c(\ell)|| > \texttt{TolCon}$, where `TolCon` is a tolerance for constraints violation (only for the Approach II).

The tolerances `TolFun`, `TolX` and `TolCon` should be specified by users.

A specific tolerance denoted by `SCQ` was introduced to enable one to conclude about the fulfilment of the Slater CQ. This tolerance is specified by users apriori: this is the desired accuracy on the regularity test. When the Approach I is being used, from the condition $G(\ell^*) < \texttt{SCQ}$, one can conclude that the Slater CQ does not hold for the SDP problem (1). When the Approach II is being used, one can conclude that the SDP problem (1) does not satisfy the Slater CQ if $H(y^*) < \texttt{SCQ}$.

6 Testing Regularity: Numerical Experiments

6.1 Description of the Experiments

Our experiments were run on a computer with an Intel Core i7-2630QM processor CPU@2.0 GHz, with Windows 7 (64 bits) and 12 GB RAM, using MATLAB (v.7.12 R2011a). The implementation of the presented procedure handles block diagonal matrices and the SDP problems should be in dat-s format. For the numerical tests we have used problems from the literature and from the SDPLIB suite, a collection of 92 Linear SDP test problems, provided by Brian Borchers [3]. In this section, we present several numerical experiments using problems from a collection of 50 SDP problems found in the literature and also instances from SDPLIB, a SDP data base containing problems ranging in size from 6 variables and 13 constraints up to 7000 variables and 7000 constraints. The problems are

drawn from a variety of applications, such as truss topology design, control systems engineering and relaxations of combinatorial optimization problems. Due to the limited computational resources, we were only able to test 26 small-scale problems from SDPLIB. The procedure proposed in this paper does not check the feasibility of the SDP problems, since it works under the assumption that their feasible sets are nonempty. Notice that all the tested SDP problems from SDPLIB are feasible, and thus, it is not required to verify their feasibility. The test problems collected from the literature are feasible and were constructed or adapted from [3, 7, 9–11, 13, 16, 17, 19–21, 23–26, 28, 30, 31, 34].

When the Approach I was applied in the procedure to check regularity of a SDP problem, the solver `lsqnonlin` with the Levenberg-Marquardt algorithm was used. When the Approach II was applied, the solver `fmincon` with the interior point algorithm was used.

We have chosen empirically the tolerances for stopping criteria. In our computational experiments, we set the tolerances `TolFun`, `TolX` and `TolCon` to 10^{-8} for all the tests. The numerical experiments have shown that for the used tolerances, the numerical results get stabilized, i.e., for termination tolerances less than 10^{-8} all the solvers stop at the same point. Therefore, the value 10^{-8} is considered to be safe to conclude about the fulfilment of the Slater CQ, although it could be more time consuming.

In our experiments on testing regularity with problems from the SDPLIB suite, we set $\texttt{SCQ} = 10^{-4}$. It may be emphasized that the value of the parameter $\texttt{SCQ} = 10^{-4}$ was considered to be a reasonable parameter in practice when large scale problems are involved. If we force a smaller value for this parameter the solvers are less efficient, since the computation time of the experiments increases in a non acceptable way.

6.2 Regularity Results

We used our procedures for testing the regularity of 50 SDP problems from literature (see Table 1) and for 26 instances from SDPLIB (see Table 3). The first columns of the tables contain the instance's name. The second and third columns contain the number of variables, n, and the dimension of the constraint matrices, s, respectively. The next three columns represent the obtained results and conclusions in terms of the fulfilment of the Slater CQ using the Approach I, where the solver `lsqnonlin` is applied for solving the quadratic system (8). The last columns of the table contain the results and conclusions about the fulfilment of the Slater CQ using the Approach II, where the solver `fmincon` is applied for solving the quadratic system. The lack of results in Table 3 means that the procedure was not able to fulfil the test, because the running time increases in an unacceptable way.

Considering the results reported in Table 1, we can see that for the feasible problems collected from the literature, both numerical approaches, Approach I and

Table 1 Numerical results about the fulfilment of the Slater CQ in SDP using the Approach I and Approach II on problems from literature (computation time is in seconds)

			`lsqnonlin`		Slater	`fmincon`		Slater
Problem	n	s	$G(\ell^*)$	Time	CQ	$H(y^*)$	Time	CQ
example3isa	2	2	$1.1363e-25$	0.036052	No	$6.9483e-15$	0.155928	No
FreundSun	2	3	$5.8581e-1$	0.070417	Yes	$5.8581e-1$	0.172609	Yes
helmberg1	2	3	$5.4540e-24$	0.060488	No	$1.5165e-13$	0.180284	No
jansson	3	3	$5.5140e-24$	0.0.065630	No	$6.3023e-14$	0.395749	No
Jansson1	3	3	$3.2587e-2$	0.138249	Yes	$3.2587e-2$	0.105394	Yes
Jansson2	3	3	$3.2619e-3$	0.073290	Yes	$3.2619e-3$	0.143587	Yes
Jansson3	3	3	$3.3027e-1$	0.060724	Yes	$3.3027e-1$	0.097528	Yes
Jansson4	3	3	$1.4601e-1$	0.069686	Yes	$1.4601e-1$	0.093127	Yes
Jansson5	3	3	$3.2127e-1$	0.071036	Yes	$3.2127e-1$	0.099324	Yes
kojima1SDP2006	2	4	$9.5204e-1$	0.223964	Yes	$9.5204e-1$	0.188890	Yes
kojimaSDP2006	4	3	$2.5000e-1$	0.023337	Yes	$2.5000e-1$	0.109484	Yes
K-Tn1	1	2	$2.5841e-23$	0.023267	No	$1.3029e-14$	0.118685	No
K-Tn2	2	2	$3.4530e-24$	0.044250	No	$5.5507e-15$	0.216425	No
K-Tn3	3	2	$5.8301e-23$	0.039288	No	$6.3543e-15$	0.124848	No
K-Tn4	4	2	$1.8636e-23$	0.041151	No	$3.8451e-15$	0.159502	No
K-Tn5	5	2	$1.8835e-22$	0.042491	No	$1.1144e-15$	0.190099	No
K-Tn6	6	2	$3.8912e-21$	0.043691	No	$1.7715e-15$	0.183893	No
K-Tn7	7	2	$3.6967e-22$	0.048549	No	$1.8502e-15$	0.270957	No
K-Tn8	8	2	$3.4387e-21$	0.050035	No	$6.6118e-16$	0.214851	No
K-Tn9	9	2	$6.4710e-22$	0.055816	No	$2.1353e-16$	0.274397	No
K-Tn10	10	2	$1.5483e-20$	0.058125	No	$4.1765e-17$	0.340125	No
LuoSturmZhang	2	3	$1.0255e-23$	0.066404	No	$7.3305e-15$	0.404231	No
Mitchell2004	2	3	$5.0000e-1$	0.045751	Yes	$5.0000e-1$	0.423425	Yes
pataki1	1	2	$4.2345e-24$	0.025313	No	$1.6671e-14$	0.187986	No
pataki1alpha1	1	2	$9.1435e-24$	0.024894	No	$4.1820e-12$	0.048927	No
pataki1alpha2	1	2	$5.2021e-23$	0.024497	No	$2.8976e-14$	0.189384	No
pataki1alpha3	1	2	$7.6752e-25$	0.024970	No	$2.6538e-17$	0.066301	No
pataki1alpha4	1	2	$2.1773e-25$	0.031795	No	$1.4323e-14$	0.183482	No
pataki1alpha5	1	2	$1.5810e-22$	0.026075	No	$7.3292e-15$	0.213028	No
pataki1alpha-1	1	2	$6.0371e-25$	0.030338	No	$1.7755e-14$	0.148022	No
pataki1alpha-2	1	2	$5.5582e-25$	0.037817	No	$9.7994e-15$	0.145855	No
pataki1alpha-3	1	2	$4.2037e-26$	0.038157	No	$4.7202e-15$	0.151887	No
pataki1alpha-4	1	2	$2.3310e-26$	0.034479	No	$9.2661e-15$	0.227901	No
pataki1alpha-5	1	2	$2.7032e-23$	0.024885	No	$1.9230e-14$	0.195679	No
pataki2	2	3	$1.0996e-23$	0.068269	No	$4.6525e-15$	0.443608	No
pataki2.-1	2	3	$2.1120e-22$	0.060252	No	$1.0909e-14$	0.433127	No
pataki2.32	2	3	$5.3683e-21$	0.058523	No	$1.9661e-14$	0.284285	No
pataki2.33	2	3	$1.2313e-22$	0.065221	No	$1.5303e-14$	0.392578	No
polik1	1	2	$4.3130e-24$	0.024449	No	$4.0894e-15$	0.142294	No
polik2	2	3	$2.8493e-23$	0.060726	No	$3.2616e-14$	0.411495	No
polik3	1	2	$1.6265e-26$	0.026367	No	$5.6213e-15$	0.163265	No
polik4	2	3	$3.4310e-24$	0.063227	No	$9.2481e-15$	0.364975	No
polik5	4	3	$1.8360e-24$	0.078022	No	$1.7116e-14$	0.575205	No
polik6	2	2	$7.3225e-25$	0.040701	No	$2.6444e-14$	0.146335	No
polik7	2	2	$1.9659e-24$	0.028269	No	$1.7242e-16$	0.134994	No
polik8	2	2	$3.3333e-1$	0.010292	Yes	$3.3333e-1$	0.068612	Yes
SturmZhang	3	5	$6.4972e-2$	0.442867	Yes	$6.4972e-2$	0.326788	Yes
Todd	2	2	$3.1557e-23$	0.029669	No	$2.2196e-15$	0.140185	No
VandBoyd1	2	3	$1.5896e-23$	0.025929	No	$5.1099e-15$	0.137883	No
YumingZhang1995	4	4	$1.4186e-22$	0.125032	No	$8.9888e-17$	0.407860	No

Approach II, performed very well and the obtained results do coincide for the two proposed approaches.

From Table 1 we can see that 39 of the 50 tested problems do not satisfy the Slater CQ, while 11 do satisfy. It is interesting to observe that all these 50 problems were correctly classified in terms of the fulfilment of the Slater CQ, since it is easy to see that 11 problems for which the Slater CQ holds, have a nonempty interior feasible set.

For these 50 problems we have applied our procedure for testing regularity and, in the cases where the Slater CQ did not hold, computed the irregularity degrees. This irregularity degree is the dimension s^* of the subspace of immobile indices of the given SDP problem. Table 2 displays the results using all the iterations of the DIIS algorithm for the 50 problems collected from the literature.

The first column in Table 2 contains the instance's name. The next two columns contain the number of variables, n, and the dimension of the constraint matrices, s. The remaining columns contain the dimension of the immobile index subspace, s^*, and the computation time. Notice that if $s^* = 0$, then the Slater CQ holds.

Observing Table 2, we can conclude that the dimension of the immobile index subspace not only provides information on the regularity of the SDP problem, but also the degree of irregularity of the problem, in the case of nonregular problems. Notice that all the problems that do not satisfy the Slater CQ have minimal irregularity degree. The only exception is the problem *YumingZhang1995* that has the irregularity degree equal to 3, which is less than $s = 4$.

Based on these experiments, one can conclude that our procedure based on the DIIS algorithm is an efficient procedure to verify if a given small-scale SDP problem satisfies the Slater CQ.

In the following, we present the numerical results on testing the regularity of SDP problems using instances from the SDPLIB suite.

Recall that the Slater CQ is satisfied when system (8) is not consistent, and in this case, we may conclude that the dimension of the immobile index subspace is $s^* = 0$.

First of all, let us observe that the computation time is much better when we used Approach II based on the `fmincon` solver for solving the quadratic system (8). The only exceptions are for problems *truss1*, *truss3* and *truss4*.

Considering the results displayed in Table 3, we can observe that for 19 of the 26 tested problems the results using both numerical approaches do coincide. Our tests show that using any of the approaches to check the fulfilment of the Slater CQ, 11 SDP tested problems satisfy the Slater CQ, while 8 do not satisfy. Notice that for 7 SDP problems, the procedure was not able to solve the system (8) using the Approach I, since the computation time have increased in an unaffordable way.

It is worth mentioning that the presented numerical procedures may return different results in terms of conclusions. Therefore, when using our numerical procedure to check the fulfilment of the Slater CQ it is recommended to run the procedure for different starting points in order to increase the degree of certainty on the obtained result/conclusion.

Table 2 Numerical results using the DIIS algorithm on problems collected from literature (computation time is in seconds)

			lsqnonlin			Slater	fmincon			Slater
Problem	*n*	*s*	s^*	Iter	Time	CQ	s^*	Iter	Time	CQ
example3isa	2	2	1	2	0.226202	No	1	2	0.539775	No
FreundSun	2	3	0	1	0.146649	Yes	0	1	0.259852	Yes
helmberg1	2	3	1	2	0.178968	No	1	2	0.658152	No
jansson	3	3	1	2	0.164909	No	1	2	0.840702	No
Jansson1	3	3	0	1	0.244887	Yes	0	1	0.170550	Yes
Jansson2	3	3	0	1	0.109592	Yes	0	1	0.193711	Yes
Jansson3	3	3	0	1	0.525396	Yes	0	1	0.279601	Yes
Jansson4	3	3	0	1	0.114246	Yes	0	1	0.127950	Yes
Jansson5	3	3	0	1	0.101950	Yes	0	1	0.144949	Yes
kojima1SDP2006	2	4	0	1	0.387002	Yes	0	1	0.303760	Yes
kojimaSDP2006	4	3	0	1	0.032463	Yes	0	1	0.174287	Yes
K-Tn1	1	2	1	2	0.072533	No	1	2	0.447973	No
K-Tn2	2	2	1	2	0.093047	No	1	2	0.447500	No
K-Tn3	3	2	1	2	0.083016	No	1	2	0.538221	No
K-Tn4	4	2	1	2	0.097496	No	1	2	0.637090	No
K-Tn5	5	2	1	2	0.120930	No	1	2	0.670066	No
K-Tn6	6	2	1	2	0.140890	No	1	2	0.762693	No
K-Tn7	7	2	1	2	0.142380	No	1	2	0.766286	No
K-Tn8	8	2	1	2	0.148806	No	1	2	0.877893	No
K-Tn9	9	2	1	2	0.145744	No	1	2	0.873110	No
K-Tn10	10	2	1	2	0.135638	No	1	2	0.907883	No
LuoSturmZhang	2	3	1	2	0.162486	No	1	2	0.518502	No
Mitchell2004	2	3	0	1	0.070062	Yes	0	1	0.634781	Yes
pataki1	1	2	1	2	0.121993	No	1	2	0.433417	No
pataki1alpha1	1	2	1	2	0.100748	No	1	2	0.806680	No
pataki1alpha2	1	2	1	2	0.129894	No	1	2	0.497163	No
pataki1alpha3	1	2	1	2	0.071018	No	1	2	0.458166	No
pataki1alpha4	1	2	1	2	0.127574	No	1	2	0.428617	No
pataki1alpha5	1	2	1	2	0.180141	No	1	2	0.479815	No
pataki1alpha-1	1	2	1	2	0.170281	No	1	2	0.419834	No
pataki1alpha-2	1	2	1	2	0.099228	No	1	2	0.525398	No
pataki1alpha-3	1	2	1	2	0.074760	No	1	2	0.555413	No
pataki1alpha-4	1	2	1	2	0.099995	No	1	2	0.824851	No
pataki1alpha-5	1	2	1	2	0.131180	No	1	2	0.445150	No
pataki2	2	3	1	2	0.177333	No	1	2	0.523884	No
pataki2.-1	2	3	1	2	0.201498	No	1	2	0.620947	No
pataki2.32	2	3	1	2	0.227159	No	1	2	0.559715	No
pataki2.33	2	3	1	2	0.191399	No	1	2	0.667981	No
polik1	1	2	1	2	0.091698	No	1	2	0.392195	No
polik2	2	3	1	2	0.165904	No	1	2	0.611956	No
polik3	1	2	1	2	0.084446	No	1	2	0.431411	No
polik4	2	3	1	2	0.151859	No	1	2	0.642326	No
polik5	4	3	1	2	0.319945	No	1	2	1.059643	No
polik6	2	2	1	2	0.097512	No	1	2	0.527270	No
polik7	2	2	1	2	0.114855	No	1	2	0.477063	No
polik8	2	2	0	1	0.020385	Yes	0	1	0.110016	Yes
SturmZhang	3	5	0	1	0.507338	Yes	0	1	0.380577	Yes
Todd	2	2	1	2	0.089365	No	1	2	0.561483	No
VandBoyd1	2	3	1	2	0.086054	No	1	2	0.422023	No
YumingZhang1995	4	4	3	2	0.301380	No	3	2	1.424340	No

Table 3 Numerical results about the fulfilment of the Slater CQ in SDP using the Approach I and Approach II using problems from SDPLIB (computation time is in seconds)

			`lsqnonlin`		Slater	`fmincon`		Slater
Problem	n	s	$G(\ell^*)$	Time	CQ	$H(y^*)$	Time	CQ
control1	21	15	$3.3333e-1$	939.7	Yes	$3.3333e-1$	50.5	Yes
control2	66	30	$3.3333e-1$	42405.0	Yes	$3.3333e-1$	1287.2	Yes
control3	136	45	$3.3333e-1$	259224.2	Yes	$3.3333e-1$	12657.3	Yes
hinf1	13	14	$3.2289e-3$	461.9	Yes	$3.2289e-3$	26.1	Yes
hinf2	13	16	$1.1797e-6$	816.8	No	$1.1797e-6$	230.6	No
hinf3	13	16	$1.7767e-7$	5301.1	No	$1.7767e-7$	135.0	No
hinf4	13	16	$4.4624e-5$	13675.2	No	$4.4622e-5$	86.2	No
hinf5	13	16	$2.7754e-9$	22167.8	No	$2.7754e-9$	221.6	No
hinf6	13	16	$6.9476e-9$	101429.0	No	$6.9476e-9$	122.3	No
hinf7	13	16	$1.6784e-10$	23130.5	No	$1.6786e-10$	202.4	No
hinf8	13	16	$1.2890e-7$	1825.3	No	$1.2890e-7$	151.5	No
hinf9	13	16	$2.2907e-11$	1562.2	No	$2.2022e-11$	827.9	No
hinf10	21	18				$4.2813e-5$	161.1	No
hinf11	31	22				$3.5089e-4$	375.0	Yes
hinf12	43	24				$1.8611e-5$	1038.6	No
hinf13	57	30				$4.3625e-9$	5902.5	No
hinf14	73	34				$2.3778e-6$	10123.4	No
hinf15	91	37				$7.9113e-10$	55173.8	No
qap5	136	26	$4.7906e-1$	118064.7	Yes	$4.7904e-1$	3508.2	Yes
qap6	229	37	$4.8191e-1$	432101.6	Yes	$4.8191e-1$	21827.1	Yes
qap7	358	50	$4.8414e-1$	433254.7	Yes	$4.8048e-1$	57981.6	Yes
qap8	529	65	$4.8192e-1$	518749.0	Yes	$4.8193e-1$	121362.1	Yes
theta1	104	50				$4.9999e-1$	18840.7	Yes
truss1	6	13	$1.4285e-1$	4.0	Yes	$1.4284e-1$	7.5	Yes
truss3	27	31	$3.2258e-2$	178.5	Yes	$3.2256e-2$	408.5	Yes
truss4	12	19	$7.6923e-2$	20.0	Yes	$7.6923e-2$	30.6	Yes

6.3 Comparison of Regularity Results

For the comparison of tests of regularity in terms of the Slater CQ with that of well-posedness, 26 test problems from the SDPLIB suite were chosen. For these problems we checked regularity in terms of the fulfilment of the Slater CQ using the DIIS algorithm and applying the Approach II, that showed to be the most efficient numerical procedure on solving the quadratic system (8). The results of testing the same problems in terms of well-posedness are presented in [8] and [13]. Notice that the results reported in [8] and [13] include all the feasible problems from SDPLIB, since they were obtained using more powerful computational resources.

The results of testing the 26 problems using the numerical procedure suggested in this paper are displayed in Table 4. The first column of the table contains the

Table 4 Numerical results on testing regularity: using the DIIS algorithm (the SDP problem satisfies the Slater CQ if $s^* = 0$), the lower and upper bounds of the Renegar condition number from [8] and the upper bound of the optimal value from [13] (if C or $\bar{p}^*$ is finite, then the problem is well-posed)

				C		
Problem	n	s	s^*	Lower bound	Upper bound	$\bar{p}^*$
control1	21	15	0	8.3×10^5	1.8×10^6	-1.7782×10^1
control2	66	30	0	3.9×10^6	1.3×10^7	-8.2909×10^0
control3	136	45	0	2.0×10^6	1.2×10^7	-1.3615×10^1
hinf1	13	14	0	∞	∞	∞
hinf2	13	16	16	3.5×10^5	5.6×10^5	-7.1598×10^0
hinf3	13	16	16	∞	∞	∞
hinf4	13	16	16	∞	∞	∞
hinf5	13	16	16	∞	∞	∞
hinf6	13	16	16	∞	∞	∞
hinf7	13	16	16	∞	∞	∞
hinf8	13	16	16	∞	∞	∞
hinf9	13	16	16	2.0×10^7	3.6×10^7	∞
hinf10	21	18	18	∞	∞	∞
hinf11	31	22	0	∞	∞	∞
hinf12	43	24	24	∞	∞	∞
hinf13	57	30	30	∞	∞	∞
hinf14	73	34	34	∞	∞	∞
hinf15	91	37	37	∞	∞	∞
qap5	136	26	0	∞	∞	∞
qap6	229	37	0	∞	∞	∞
qap7	358	50	0	∞	∞	∞
qap8	529	65	0	∞	∞	∞
theta1	104	50	0	2.0×10^2	2.1×10^2	-2.3000×10^1
truss1	6	13	0	2.2×10^2	3.0×10^2	9.0000×10^0
truss3	27	31	0	7.4×10^2	1.9×10^3	9.1100×10^0
truss4	12	19	0	3.6×10^2	7.7×10^2	9.0100×10^0

instance's name used in SDPLIB data base; the next two columns refer to the number of variables, n, and the dimension of the constraint matrices, s. The next column represents the results of the regularity tests that find the dimension of the immobile index subspace, s^*. If $s^* = 0$, then the problem satisfies the Slater CQ. Column 5 contains the lower and upper bounds of the condition number C reported in [8] and the last column presents the upper bound for the primal objective function from [13].

From Table 4 we see that 13 from 26 tested problems are regular, i.e., their constraints satisfy the Slater CQ. Moreover, the tests provide valuable information about the irregularity degree for a nonregular SDP problem, i.e., when the Slater CQ

Table 5 Regularity in terms of the fulfilment of Slater's CQ and well-posedness according to [8]

	Regularity (Slater CQ)	
	Regular	Nonregular
Well-posed	7	2
Ill-posed	6	11

Table 6 Regularity in terms of the fulfilment of Slater's CQ and well-posedness according to [13]

	Regularity (Slater CQ)	
	Regular	Nonregular
Well-posed	7	1
Ill-posed	6	12

fails to hold. Notice that for all the tested problems in Table 4 for which the Slater CQ fails have maximal irregularity degree.

Tables 5 and 6 compare the results of testing SDP problems in terms of their regularity: the fulfilment of the Slater CQ and well-posedness.

In Table 5, the lines correspond to well-posed and ill-posed problems according to the test from [8] and the columns correspond to regular and nonregular problems in terms of the Slater CQ. On the intersection we have the number of problems that satisfy both corresponding conditions. Table 6 is constructed in a similar way, but the lines correspond to the number of the well-posed and ill-posed problems classified on the basis of the experiments in [13]. From Table 5 we can see that 18 from the tested problems satisfy the Slater CQ and are well-posed, or do not satisfy the Slater CQ and are ill-posed, simultaneously, and 6 of the ill-posed problems do satisfy the Slater CQ. The only exception is problem *hinf9* that is nonregular in terms of the fulfilment of the Slater CQ and well-posed according to [8]. This contradiction to Lemma 1 can be explained by the fact that the numerical procedures are based on approximated calculus and may be not precise. Comparing our regularity results in terms of the Slater CQ with those from [13] w.r.t. ill-posedness, regarding Table 6 we conclude that for 19 problems these results coincide, i.e., the problems satisfy the Slater CQ and are well-posed, or do not satisfy the Slater CQ and are ill-posed, simultaneously.

Notice that the numerical results of well-posedness obtained in [8] and in [13] do not coincide: problem *hinf9* is well-posed according to [8] and ill-posed according to [13]. This can be connected with the fact that nevertheless the condition number C is finite, it is rather big and the problem is close to be ill-posed, and it may also be due to the tests were performed in nonexact arithmetic and/or with different numerical procedures.

Finally, notice that in [13], it is reported that problem *hinf8* is well-posed, although the results presented in [13] (and also in [8]) have shown that this problem is ill-posed. Our numerical tests show that this problem is nonregular in terms of the fulfilment of the Slater CQ.

Therefore, the numerical experiences have showed that the DIIS algorithm can be efficiently used to study the regularity of SDP problems in terms of the Slater CQ.

The comparison of these tests with those from [8] and [13] confirm the conclusions about relationship between the regularity notions in SDP.

6.4 Conclusions and Future Work

In this paper we have presented a simple numerical procedure to test regularity on SDP problems in terms of the Slater CQ, within a desired tolerance. This procedure is based on the DIIS algorithm, which revealed to be very efficient on checking regularity on SDP. Our procedure can use two numerical approaches: the first, based on a least-squares problem and a second one, based on a constrained nonlinear problem. Both approaches were tested on problems collected from literature and also from SDPLIB and a comparative analysis was performed.

On the basis of our numerical experiments, we can conclude that: our procedure is efficient on checking regularity of SDP's; it is important to introduce a unified treatment of regularity for SDP problems, have numerical tools to verify regularity, and establish clear relationship between different notions of regularity; the cross-check of different numerical approaches is important, since it permits to increase the reliability on the results.

In the future, we plan to improve the procedure described in the paper implementing a pre-step to verify feasibility of SDP problems, since the DIIS algorithm can only be applied on feasible SDP problems. Moreover, we are going to provide more extensive regularity tests on SDP and SIP problems, and compare them with other available results.

Acknowledgements The author would like to thank the anonymous referee for the valuable comments that have helped to improve the paper. This work was supported by Portuguese funds through the CIDMA – Center for Research and Development in Mathematics and Applications (University of Aveiro), and the Portuguese Foundation for Science and Technology ("FCT – Fundação para a Ciência e a Tecnologia"), within project UID/MAT/04106/2013.

References

1. Anjos, M.F., Lasserre, J.B. (eds.): Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications. International Series in Operational Research and Management Science, vol. 166. Springer, New York (2012)
2. Bonnans, J.F., Shapiro, A.: Perturbation Analysis of Optimization Problems. Springer, New York (2000)
3. Borchers, B.: SDPLIB 1.2, a library of semidefinite programming test problems. Optim. Methods Softw. **11**(1), 683–690 (1999)
4. Cheung, Y., Schurr, S., Wolkowicz, H.: Preprocessing and Reduction for Degenerate Semidefinite Programs. Research Report CORR 2011-02. http://www.optimization-online.org/DB_FILE/2011/02/2929.pdf (2013). Revised in January 2013

5. Dontchev, A.L., Zolezzi, T.: Well-Posed Optimization Problems. Lecture Notes in Mathematics, vol. 1543. Springer, Berlin (1993)
6. Freund, R.M.: Complexity of an Algorithm for Finding an Approximate Solution of a Semi-Definite Program, with no Regularity Condition (1995). Technical Report OR 302-94, Op. Research Center, MIT, Revised in December 1995
7. Freund, R.M., Sun, J.: Semidefinite Programming I: Introduction and minimization of polynomials, System Optimization. Available at http://www.myoops.org/cocw/mit/NR/rdonlyres/Sloan-School-of-Management/15-094Systems-Optimization--Models-and-ComputationSpring2002/1B59FD11-A822-4C80-9301-47B127500648/0/lecture22.pdf (2002)
8. Freund, R.M., Ordóñez, F., Toh, K.C.: Behavioral Measures and their Correlation with IPM Iteration Counts on Semi-Definite Programming Problems. Math. Program. **109**(2), 445–475 (2007). Springer, New York
9. Fujisawa, K., Futakata, Y., Kojima, M., Matsuyama, S., Nakamura, S., Nakata, K., Yamashita, M.: SDPA-M (SemiDefinite Programming Algorithm in MATLAB) User's Manual-V6.2.0, Series B: OR Department of Mathematical and Computing Sciences (2005)
10. Gärtner, B., Matoušek, J.: Interior Point Methods, Approximation Algorithms and Semidefinite Programming. Available at http://www.ti.inf.ethz.ch/ew/courses/ApproxSDP09/ (2009)
11. Helmberg, C.: Semidefinite Programming for Combinatorial Optimization. ZIB Report, Berlin (2000)
12. Hernández-Jiménez, B., Rojas-Medar, M.A., Osuna-Gómez, R., Beato-Moreno, A.: Generalized convexity in non-regular programming problems with inequality-type constraints. J. Math. Anal. Appl. **352**(2), 604–613 (2009)
13. Jansson, C., Chaykin, D., Keil, C.: Rigorous error bounds for the optimal value in semidefinite programming. SIAM J. Numer. Anal. archive **46**(1), 180–200 (2007)
14. Klatte, D.: First order constraint qualifications. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, 2nd edn, pp. 1055–1060. Springer, US (2009)
15. Klerk, E. de: Aspects of Semidefinite Programming – Interior Point Algorithms and Selected Applications. Applied Optimization, vol. 65. Kluwer, Boston (2004)
16. Kojima, M.: Introduction to Semidefinite Programs (Semidefinite Programming and Its Application), Institute for Mathematical Sciences National University of Singapore (2006)
17. Kolman, B., Beck, R.E.: Elementary Linear Programming with Applications, 2nd edn, Academic Press, San Diego (1995)
18. Konsulova, A.S., Revalski, J.P.: Constrained convex optimization problems – well-posedness and stability. Numer. Funct. Anal. Optim. **15**(7–8), 889–907 (1994)
19. Kostyukova, O.I., Tchemisova, T.V.: Optimality criterion without constraint qualification for linear semidefinite problems. J. Math. Sci. **182**(2), 126–143 (2012)
20. Luo, Z., Sturm, J., Zhang, S.: Duality results for conic convex programming, Econometric Institute Report No. 9719/A (1997)
21. Mitchell, J., Krishnan, K.: A unifying framework for several cutting plane methods for semidefinite programming, Technical Report, Department of Computational and Applied Mathematics, Rice University (2003)
22. Moré, J.J.: The Levenberg-Marquardt algorithm: implementation and theory. In: Watson, G.A. (ed.) Numerical Analysis. Lecture Notes in Mathematics, vol. 630, pp. 105–116. Springer, Berlin/Heidelberg (1977)
23. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
24. Pataki, G.: Bad semidefinite programs: they all look the same, Technical report, Department of Operations Research, University of North Carolina (2011)
25. Pedregal, P.: Introduction to Optimization. Springer, New York (2004)
26. Polik, I.: Semidefinite programming Feasibility and duality. Available at http://imre.polik.net/wp-content/uploads/IE496/POLIK_IE496_04_duality.pdf (2009)
27. Renegar, J.: Some perturbation-theory for linear-programming. Math. Program. **65**(1), 73–91 (1994)

28. Sturm, J., Zhang, S.: On sensitivity of central solutions in semidefinite programming. Math. Program. **90**(2), 205–227 (1998). Springer
29. Tikhonov, A.N., Arsenin, V.Y.: Solutions of ill-posed problems. John Wiley and Sons, New York (1977)
30. Todd, M.J.: Semidefinite optimization. Acta Numer. **10**, 515–560 (2001). Cambridge University Press
31. Vandenberghe, L., Boyd, S.: Semidefinite Programming. SIAM Rev. **38**(1), 49–95 (1996)
32. Wolkowicz, H.: Duality for semidefinite programming. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, 2nd edn, pp. 811–814. Springer, US (2009)
33. Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming: Theory, Algorithms, and Applications. Kluwer, Boston (2000)
34. Zhang, Y.: Semidefinite Programming, Lecture 2. Available at http://rutcor.rutgers.edu/~alizadeh/CLASSES/95sprSDP/NOTES/lecture2.ps (1995)

Decompositions and a Matheuristic for a Forest Harvest Scheduling Problem

Isabel Martins, Filipe Alvelos, and Miguel Constantino

Abstract In this paper, we describe four decomposition models and a matheuristic based on column generation for the forest harvest scheduling problems subject to maximum area restrictions. Each of the four decomposition models can be seen as a Dantzig-Wolfe decomposition of the so-called bucket formulation (compact mixed integer program), in two cases with additional constraints on the connectivity of the buckets. The matheuristic is based on one of the decomposition models (the $\mathscr{S}$-knapsack-and-clique decomposition) and relies on the interaction of column generation with a general purpose mixed integer programming solver. We compare the quality of the solutions obtained for benchmark instances with the bucket formulation and with applying column generation and solving the *integer* restricted master problem (MipHeur) for the same time limit. We concluded that the proposed matheuristic provides, in general, better solutions than both the other approaches for small and medium instances, while, for large instances, the MipHeur approach outperformed the other two.

I. Martins (✉)
Centro de Matemática, Aplicações Fundamentais e Investigação Operacional/Departamento de Ciências e Engenharia de Biossistemas, Instituto Superior de Agronomia, 1349-017 Lisboa, Portugal
e-mail: isabelinha@isa.ulisboa.pt

F. Alvelos
Centro Algoritmi/Departamento de Produção e Sistemas, Universidade do Minho, 4710-057 Braga, Portugal
e-mail: falvelos@dps.uminho.pt

M. Constantino
Centro de Matemática, Aplicações Fundamentais e Investigação Operacional/Departamento de Estatística e Investigação Operacional, Faculdade de Ciências de Lisboa, 1749-016 Lisboa, Portugal
e-mail: miguel.constantino@fc.ul.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_14

1 Introduction

Forest harvesting for timber production causes negative environmental impacts, primarily habitat alteration and loss of biodiversity, soil, water quality and scenic beauty. A common practice in many countries to reduce these impacts has been to restrict the areas of clearcuts. Addressing these constraints has led to an evolution of model approaches that support forest management. The most recent approach, the so-called area restriction model (ARM), lets the formulation itself suggest stand aggregation when the sum of the areas does not violate the maximum clearcut area. Three main basic integer programming models for the ARM have been described in the literature: the path formulation, with an exponential number of constraints, the cluster formulation, with an exponential number of variables, and the bucket formulation, with a polynomial number of variables and constraints (for a survey of integer programming approaches to solving the ARM, we refer those interested to [6]).

The harvest scheduling problem that we shall consider consists of selecting, for each period in the planning horizon, a set of stands to be harvested, in order to maximize the timber's net present value. The stand selection is subject to several restrictions. Maximum area restrictions impose that the area of each clearcut does not exceed the maximum allowed size. Each stand is harvested at the most once in the planning horizon, i.e. the minimum rotation in the stand is longer than the latter. Other requirement is a steady flow of harvested timber. This restriction is mainly to ensure that the industry is able to continue operating with similar levels of machine and labor utilizations.

Four Dantzig-Wolfe decompositions of the bucket formulation [3] are proposed: the $\mathscr{S}$-knapsack and the $\mathscr{S}$-knapsack-and-clique decompositions, and two similar decompositions of the bucket formulation with additional constraints on the connectivity of the buckets, the $\mathscr{R}$-knapsack and the $\mathscr{R}$-knapsack-and-clique decompositions. We establish theoretically that the LP bounds of the knapsack-and-clique decompositions are better than or equal to those of the knapsack decompositions, and that the LP bound of the $\mathscr{S}$-knapsack-and-clique decomposition is equal to that of the $\mathscr{R}$-knapsack-and-clique decomposition. According to these results and those obtained with preliminary computational tests, and since the pricing subproblem of the $\mathscr{S}$-knapsack-and-clique decomposition may be less difficult to solve than that of the $\mathscr{R}$-knapsack-and-clique decomposition, we decide to present the solution approach based on the $\mathscr{S}$-knapsack-and-clique decomposition.

For solving the decomposition model of the $\mathscr{S}$-knapsack-and-clique decomposition, we propose a matheuristic (MatHeur), a heuristic based on mathematical programming. The MatHeur is closely related with the general framework "Metaheuristic search by column generation", *SearchCol* for short, designed to solve integer programming and combinatorial optimization problems with a decomposable structure [1].

A SearchCol algorithm has three main steps which are executed in cycle: (i) column generation (CG) is applied (possibly with additional constraints fixing

subproblem vaiables – named *perturbations*), (ii) a search is conducted in a space defined by the subproblem solutions provided by CG, and (iii) the set of subproblem variables currently fixed in CG is updated. The difference between the proposed approach and SearchCol is that in SearhcCol, a metaheuristic is used in step (ii) instead of a general purpose mixed integer programming solver.

SearchCol defines several ways of defining perturbations [1]. In the proposed MatHeur we explore the definition of perturbations based on scoring the subproblem solutions according to information from the incumbent, previous CGs, and previous searches. We also explore a intensification strategy consisting in, for each even iteration, defining a restricted search space made of the subproblem solutions in the incumbent, null, generated last time CG was solved, and that were included more times in solutions obtained in previous searches.

We describe the harvest scheduling problem and the bucket formulation in Sect. 2, and the decompositions in Sect. 3. The results of the dominance relationships between the bounds of the bucket formulation and the decompositions are presented in Sect. 3. In Sect. 4, we describe the basis of the proposed MatHeur. In Sect. 5, we report on computational experience as to the efficiency of the MatHeur. Our tests involved both real and hypothetical forests ranging from 45 to 2945 polygons and used temporal horizons ranging from three to twelve periods. We compare the quality of the solutions obtained for these instances with the bucket formulation for the same time limit. In the last section, we present our conclusions.

2 Problem Definition and the Bucket Compact Model

The addressed forest harvest scheduling problem consists in determining which stands should be harvested in each period during a planning horizon. The area of each contiguous set of harvested stands cannot exceed the maximum allowed size, and the variation on the volume of timber harvested in each period cannot exceed a given degree of fluctuation from the harvest level in the previous period. Each stand can only be harvested once. The objective is to maximize the total profit (net present value) defined as the sum of the profits generated by the harvestings during the planning horizon. We introduce the following notation:

- T – set of time periods indexed by $t = 1, \ldots, |T|$
- I – set of stands indexed by $i = 1, \ldots, |I|$
- a_i – area of stand i; $i \in I$
- A_{max} – maximum clearcut area
- p_i^t – timber net present value from stand i if it is harvested in period t; $i \in I$; $t \in T$
- v_i^t – volume of timber of stand i in period t; $i \in I$; $t \in T$
- Δ – maximum allowed variation on volume of timber harvested between two consecutive periods.

Let $G = (I, E)$ be a graph, where each stand in I is represented by a vertex and the endpoints of each edge in E correspond to two adjacent stands. Using

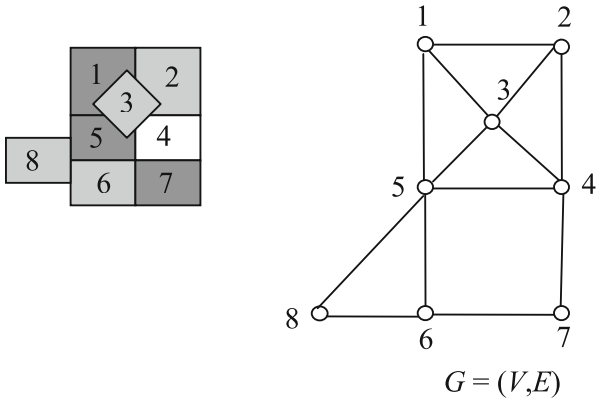


Fig. 1 Forest with eight stands and its graph representation. Stands 2, 3, 6 and 8 are harvested in period 1, stands 1, 5 and 7 in period 2, and there is no intervention in stand 4. The area of each stand is 1 ha

the definition of strong adjacency, the graph is planar, i.e. it can be drawn in a plane surface without crossing edges. Let $\mathcal{Q}$ be the set of maximal cliques of G indexed by $P \in \mathcal{Q}$. A clique is the set of nodes of a complete subgraph of the graph, which has an edge between each pair of vertices, and it is maximal if it is not contained in any other clique. Since the graph is planar there are no cliques with more than four vertices [4]. For the graph in Fig. 1, $\mathcal{Q} = \{\{1,2,3\},\{1,3,5\},\{2,3,4\},\{3,4,5\},\{4,7\},\{5,6,8\},\{6,7\}\}$. Cliques are used to ensure that each clearcut does not exceed the maximum allowed size.

We now describe a compact formulation, the bucket formulation.

As each stand is harvested once at the most, the harvested area in a forest is a set of clearcuts (maximal harvested connected regions) which do not overlap in the course of the planning horizon. Thus, each clearcut may be represented by one of its stands, for example the one with the smallest index. In this paper two stands are considered to be adjacent when both share a boundary with positive length, i.e. that is not a discrete set of points (the so-called strong adjacency [5]). According to this definition, clearcuts in Fig. 1 are, in period 1, regions $\{2,3\}$ and $\{6,8\}$ and, in period 2, $\{1,5\}$ and $\{7\}$. These clearcuts may be represented by stands 2, 6, 1 and 7, respectively.

Let us consider the empty set C_k (*bucket*) for each stand k. Assigning stands to C_k (stand k and stands $i > k$) corresponds to selecting these stands to be harvested. The model assigns stands to buckets in such a way that constraints of the maximum area, volume variation, at most one harvest per stand, are satisfied. A bucket remains empty if no stands are assigned to it. Each non empty bucket represents a feasible clearcut or a set of feasible clearcuts. The stand with the smallest index in a non empty bucket C_k is k.

We can represent the set of buckets in the forest as $\mathrm{C} = \{C_1, \ldots, C_{|I|}\}$.

The decision variables in the bucket model are therefore as follows:

$$x_i^{kt} = \begin{cases} 1 \text{ if stand } i \text{ is selected to belong to bucket } C_k \text{ in period } t \\ 0 \text{ otherwise;} \quad k \in I; t \in T; i = k, \ldots, |I| \end{cases}$$

$$w_P^{kt} = \begin{cases} 1 \text{ if at least one stand from clique } P \text{ is selected to belong to bucket } C_k \\ \quad \text{in period } t \\ 0 \text{ otherwise;} \quad k \in I; t \in T; P \in \mathcal{Q} : \max_{i \in P}\{i\} \geq k. \end{cases}$$

The model is the following:

$$\max \sum_{t \in T} \sum_{k \in I} \sum_{i=k}^{|I|} p_i^t x_i^{kt} \tag{1}$$

subject to

$$x_i^{kt} \leq w_P^{kt}; k \in I; t \in T; i \geq k; P \in \mathcal{Q} : i \in P \tag{2}$$

$$\sum_{k \leq \max_{i \in P}\{i\}} w_P^{kt} \leq 1; t \in T; P \in \mathcal{Q} \tag{3}$$

$$\sum_{i=k+1}^{|I|} a_i x_i^{kt} \leq (A_{max} - a_k) x_k^{kt}; k \in I; t \in T \tag{4}$$

$$\sum_{t \in T} \sum_{k=1}^{i} x_i^{kt} \leq 1; i \in I \tag{5}$$

$$\sum_{i \in I} v_i^t \sum_{k=1}^{i} x_i^{kt} \geq (1 - \Delta) \sum_{i \in I} v_i^{t-1} \sum_{k=1}^{i} x_i^{k,t-1}; t = 2, \ldots, |T| \tag{6}$$

$$\sum_{i \in I} v_i^t \sum_{k=1}^{i} x_i^{kt} \leq (1 + \Delta) \sum_{i \in I} v_i^{t-1} \sum_{k=1}^{i} x_i^{k,t-1}; t = 2, \ldots, |T| \tag{7}$$

$$x_i^{kt} \in \{0, 1\}; k \in I; t \in T; i = k, \ldots, |I| \tag{8}$$

$$w_P^{kt} \geq 0; k \in I; t \in T; P \in \mathcal{Q} : \max_{i \in P}\{i\} \geq k. \tag{9}$$

The objective function (1) states the management objective of maximizing the net present value of timber harvested. Constraints (2) define the relationship between variables x and w. Constraints (2) and (3) ensure that in each period every two adjacent stands are in one bucket at the most. Constraints (4) guarantee that each bucket does not exceed the maximum allowed size. Constraints (4) also state that if C_k is non-empty then C_k contains stand k, and thus the stand in C_k with the smallest index is k. Constraints (5) state that each stand is harvested at the most once in the

planning horizon. Constraints (6) and (7) allow harvested volumes in each period to range from $1 - \Delta$ to $1 + \Delta$ times the harvested volume in the previous period. The other constraints state binary and non-negativity requirements on variables. The integrality of variables x, together with constraints (2), implies the integrality of variables w in at least one optimal solution. Note that a non empty bucket is a region that might be disconnected since there are no constraints to enforce its connectivity. However, any solution with a disconnected harvested set C_k is equivalent to the solution where C_k is replaced by its clearcuts, each with an area not exceeding A_{max}. Hence there is no need to add explicit constraints in the model to enforce connectivity of the buckets.

The bucket model has $O(|I| \times |\mathscr{Q}| \times |T|)$ variables and constraints. If G is planar, the number of cliques is of the order of the number of nodes [4], so in this case the formulation has $O(|I|^2 \times |T|)$ variables and constraints. Even though the number of variables is polynomial, it can be very large for large instances. However, most variables have the value zero in any feasible solution, and this can be determined a priori [3]. Observe that if a stand is too "far" from stand k then it is not worth assigning it to bucket C_k, because the area of any connected region with both stands would exceed the maximum. Back to Fig. 1, considering $A_{max} = 2$, for $k = 1$, variables x_4^{1t}, x_6^{1t}, x_7^{1t}, x_8^{1t}, $w_{4,7}^{1t}$, $w_{6,7}^{1t}$ are null in any feasible solution. Such variables may not be considered and thus the number of variables and constraints of the model can be reduced.

We pointed out above that there is no need to enforce connectivity in the bucket formulation. Nevertheless, we can consider an alternative formulation with constraints on the connectivity of the subgraphs of G induced by sets $\{i : x_i^{kt} = 1\}, k \in I$ and $t \in T$ or, in other words, on the connectivity of the non empty buckets. We consider two Dantzig-Wolfe decompositions for the bucket model and also similar decompositions for the alternative formulation. For the sake of simplicity, the alternative formulation is not described.

The *knapsak decomposition* of the bucket model is obtained by reformulating set $\mathscr{K}$ defined by constraints (4) and (8), while the *knapsack-and-clique decomposition* correspond to the reformulation of set $\mathscr{C}$ defined by constraints (2), (4), (8) and (9). Similar decompositions of the alternative formulation are obtained by reformulating sets $\mathscr{K}$ and $\mathscr{C}$ by the same constraints and the connectivity constraints. In each case, the representation of the convex hull of the corresponding set by extreme points is used to strengthen the respective formulation. As those convex hulls have an exponential number of extreme points in general, the linear programming relaxations of the resulting models are solved by column generation. The corresponding pricing subproblems consider objective functions of reduced costs over sets $\mathscr{K}$ and $\mathscr{C}$, respectively.

3 Decomposition Models

3.1 Knapsack Decompositions

In this subsection, we propose a Dantzig-Wolfe decompositions of the bucket model by reformulating the set defined by constraints (4) and (8). A similar decomposition of the alternative model can also be presented by reformulating the set defined by these constraints and the connectivity constraints.

3.1.1 The Master Problems

Let $d = |T| \times |I| \times (|I| + 1)/2$ be the number of variables x_i^{kt} in the bucket model and $\mathscr{K} = (x \in \mathbb{R}^d)$ satisfying (4) and (8).

Observe that $\mathscr{K}$ can be decomposed into $|I| \times |T|$ sets $\mathscr{K}^{kt} = \{x \in \{0,1\}^{|I|-k+1} : \sum_{i=k+1}^{|I|} a_i x_i^{kt} \leq (A_{max} - a_k) x_k^{kt}\}$. Observe further that, for a given k, sets $\mathscr{K}^{kt}$ are identical, and that an element of a set $\mathscr{K}^{kt}$ is either the null vector or the incidence vector of a region with stands $i \geq k$, containing k and with area not greater than A_{max}. That is, $\mathscr{K}^{kt} = \{0\} \cup \{\bar{x}_S^k : S \in \mathscr{S}^k\}$ where $\mathscr{S}^k$ is the set of all regions S with stands $i \geq k$ such that $k \in S$ and the area of S is not greater than A_{max}, and $\bar{x}_S^k$ is the incidence vector of S:

$$\bar{x}_{iS}^k = \begin{cases} 1 \text{ if stand } i \text{ belongs to region } S \in \mathscr{S}^k \\ 0 \text{ otherwise.} \end{cases}$$

Now, for $t \in T$, $k \in I$ and $S \in \mathscr{S}^k$ define the variables

$$y_S^{kt} = \begin{cases} 1 \text{ if region } S \text{ is selected to be harvested in period } t \\ 0 \text{ otherwise} \end{cases}$$

and for each $k \in I$ and $t \in T$ define the variable y_0^{kt} that assumes the unitary value if none of sets of $\mathscr{S}^k$ is selected to be harvested in period t and the null value otherwise. We have $\mathscr{K}^{kt} = \{x \in \mathbb{R}^{|I|-k+1} : x = \sum_{S \in \mathscr{S}^k} \bar{x}_S^k y_S^{kt}, y_0^{kt} + \sum_{S \in \mathscr{S}^k} y_S^{kt} = 1, y_S^{kt} \in \{0,1\}, S \in \mathscr{S}^k\}$, for each $k \in I$ and $t \in T$. By incorporating this reformulation of the sets $\mathscr{K}^{kt}$ into the bucket formulation, we obtain the master problem

$$\max \sum_{t \in T} \sum_{k \in I} \sum_{S \in \mathscr{S}^k} \sum_{i=k}^{|I|} p_i^t \bar{x}_{iS}^k y_S^{kt} \tag{10}$$

subject to

$$y_0^{kt} + \sum_{S \in \mathscr{S}^k} y_S^{kt} = 1; k \in I; t \in T \tag{11}$$

$$\sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_S^{kt} \leq w_P^{kt}; k \in I; t \in T; i \geq k; P \in \mathscr{Q} : i \in P \tag{12}$$

$$\sum_{k \leq \max\{i : i \in P\}} w_P^{kt} \leq 1; t \in T; P \in \mathscr{Q} \tag{13}$$

$$\sum_{t \in T} \sum_{k=1}^{i} \sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_S^{kt} \leq 1; i \in I \tag{14}$$

$$\sum_{i \in I} v_i^t \sum_{k=1}^{i} \sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_{\mathscr{S}}^{kt} \geq (1 - \Delta) \sum_{i \in I} v_i^{t-1} \sum_{k=1}^{i} \sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_S^{k,t-1}; t = 2, \ldots, |T| \tag{15}$$

$$\sum_{i \in I} v_i^t \sum_{k=1}^{i} \sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_S^{kt} \leq (1 + \Delta) \sum_{i \in I} v_i^{t-1} \sum_{k=1}^{i} \sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_S^{k,t-1}; t = 2, \ldots, |T| \tag{16}$$

$$y_S^{kt} \in \{0, 1\}; k \in I; t \in T; S \in \mathscr{S}^k \tag{17}$$

$$y_0^{kt} \in \{0, 1\}; k \in I; t \in T \tag{18}$$

$$w_P^{kt} \geq 0; k \in I; t \in T; P \in \mathscr{Q} : \max_{i \in P}\{i\} \geq k. \tag{19}$$

We shall refer to this formulation as $\mathscr{S}$-knapsack decomposition. Note that constraints (14) imply constraints (11), so these can be removed.

Observe also that variables x in the bucket formulation and variables y in the $\mathscr{S}$-knapsack decomposition are related through the equations $x_i^{kt} = \sum_{S \in \mathscr{S}^k} \bar{x}_{iS}^k y_S^{kt}$. The linear programming relaxation of the $\mathscr{S}$-knapsack decomposition corresponds to replacing, in the bucket model, the set $\mathscr{K}$ defined by constraints (4) and (8) by its convex hull. Given that the extreme points of the set defined by the linear relaxation of constraints (4) and (8) are not necessarily integer, we may state the following:

Proposition 1 *The LP bound of the $\mathscr{S}$-knapsack decomposition is better than or equal to that of the bucket formulation.*

A similar decomposition of the alternative model, with the connectivity constraints to be included in the knapsack sets to be reformulated, can also be considered. Since now this sets are more constrained, the linear programming bounds obtained by the knapsack decomposition with connectivity are not worse than those obtained by the knapsack decomposition without connectivity.

Let $\mathscr{R}^k$ denote the set of all regions from $\mathscr{S}^k$ that are connected, for $k \in I$. We shall refer to the master problem where sets $\mathscr{S}^k$ are replaced by $\mathscr{R}^k$ as $\mathscr{R}$-knapsack decomposition. The solution set of the master problem does not change if sets $\mathscr{S}^k$ are replaced by $\mathscr{R}^k$. However, we have the following:

Proposition 2 *The LP bound of the $\mathscr{R}$-knapsack decomposition is better than or equal to that of the $\mathscr{S}$-knapsack decomposition.*

That the LP bound of the $\mathscr{R}$-knapsack decomposition is never worse than that of the $\mathscr{S}$-knapsack decomposition follows from the fact that there are more constraints in the alternative formulation defining the set which is reformulated, and the remaining constraints in the master problem are the same. Next, we will give an example that shows there are instances for which the LP bound of the $\mathscr{R}$-knapsack decomposition is strictly better.

Consider a forest with seven stands (Table 1) and let $A_{max} = 3.0$ and $\Delta = 1.0$. Optimal solutions of the linear relaxations of the $\mathscr{S}$-knapsack and $\mathscr{R}$-knapsack decompositions are the following, respectively (only the non-null values are displayed):

$y^{3,1}_{\{3\}} = 0.(3)$, $y^{7,1}_{\{7\}} = 0.(3)$, $y^{5,1}_{\{5,6\}} = 1$, $y^{1,1}_{\{1,2,3\}} = 0.(3)$, $y^{1,1}_{\{1,2,7\}} = 0.(3)$, $y^{1,1}_{\{1,3,7\}} = 0.(3)$, $y^{2,2}_{\{2\}} = 0.(3)$, $y^{4,2}_{\{4\}} = 1$ (note that set $\{1, 3, 7\}$ is disconnected),

$w^{1,1}_{\{1,2\}} = 1$, $w^{1,1}_{\{2,3\}} = 0.(6)$, $w^{3,1}_{\{2,3\}} = 0.(3)$, $w^{1,1}_{\{2,7\}} = 0.(6)$, $w^{7,1}_{\{2,7\}} = 0.(3)$, $w^{1,1}_{\{3,4\}} = 0.(6)$, $w^{3,1}_{\{3,4\}} = 0.(3)$, $w^{5,1}_{\{5,6\}} = 1$, $w^{2,2}_{\{1,2\}} = 0.(3)$, $w^{2,2}_{\{2,3\}} = 0.(3)$, $w^{2,2}_{\{2,7\}} = 0.(3)$, $w^{4,2}_{\{3,4\}} = 1.0$, $w^{4,2}_{\{4,5\}} = 1.0$,

$y^{1,2}_0 = y^{3,2}_0 = y^{4,1}_0 = y^{5,2}_0 = y^{6,2}_0 = y^{7,2}_0 = 1.0$, $y^{2,1}_0 = 0.(3)$, $y^{2,2}_0 = 0.(6)$, with the value of the objective function 165,104.1;

$y^{1,1}_{\{1,2,7\}} = 1.0$, $y^{4,1}_{\{4,5,6\}} = 1.0$, $y^{3,2}_{\{3\}} = 1.0$,

$w^{1,1}_{\{1,2\}} = w^{1,1}_{\{2,3\}} = w^{1,1}_{\{2,7\}} = w^{4,1}_{\{3,4\}} = w^{4,1}_{\{4,5\}} = w^{3,2}_{\{2,3\}} = w^{3,2}_{\{3,4\}} = 1.0$,

$y^{1,2}_0 = y^{2,2}_0 = y^{3,1}_0 = y^{4,2}_0 = y^{5,2}_0 = y^{6,2}_0 = y^{7,2}_0 = 1.0$, with the value of the objective function 161,346.1. These solutions are obtained using CPLEX 12.4 [7] as a linear programming solver.

Table 1 Instance for which the LP bound of the $\mathscr{R}$-knapsack decomposition is strictly better than that of the $\mathscr{S}$-knapsack decomposition

Stand i	a_i	p^1_i	p^2_i	v^1_i	v^2_i	Nodes adjacent to i
1	1.0	27,135.0	18,387.2	524.2	558.5	2
2	1.0	26,524.1	18,030.2	512.9	548.7	1, 3, 7
3	1.0	26,524.1	18,030.2	512.9	548.7	2, 4
4	1.0	7693.6	5789.0	263.2	341.5	3, 5
5	1.0	28,094.4	18,950.6	542.0	573.9	4, 6
6	1.0	26,934.4	18,269.8	520.5	555.3	5
7	1.0	26,934.4	18,269.8	520.5	555.3	2

3.1.2 The Pricing Subproblems

Relaxation of the binary requirement on the variables $y^{kt}_{\mathscr{S}}$ leads to the linear relaxation of the master problem. Constraints (17) and (18) are simply replaced respectively by $y^{kt}_S \geq 0$ and $y^{kt}_0 \geq 0$ because constraints (11) guarantee $y^{kt}_S \leq 1$ and $y^{kt}_0 \leq 1$. The pricing subproblem kt for each node $k \in I$ and for each period $t \in T$ consists of finding a variable y^{kt}_S, with $S \in \mathscr{S}^k$ or $S \in \mathscr{R}^k$, such that the corresponding reduced cost is maximum. The variables y^{kt}_0 and w^{kt}_P are inserted into the first restricted master problem.

Let Ω^{kt}, α^{kt}_{iP}, β^t_P, θ_i, μ^t and ν^t denote the dual variables associated with constraints (11), (12), (13), (14), (15) and (16) of the linear relaxation of the master problem and Ω^{kt*}, α^{kt*}_{iP}, β^{t*}_P, θ^*_i, μ^{t*} and ν^{t*} assume an optimal dual solution of the linear relaxation of a restricted master problem. By definition of reduced cost, the objective function of the pricing subproblem kt is given by

$$\max_{S \in \mathscr{SR}} \{\sum_{i \in S} \epsilon^{t*}_i\} + \Omega^{kt*}$$

where $\mathscr{SR} = \mathscr{S}^k$ or $\mathscr{SR} = \mathscr{R}^k$ and

$$\epsilon^{1*}_i = p^1_i + \theta^*_i - \mu^{2*}(1-\Delta)v^1_i - \nu^{2*}(1+\Delta)v^1_i + \sum_{\substack{P \in \mathscr{Q}: \\ i \in P}} \alpha^{k1*}_{iP}$$

$$\epsilon^{t*}_i = p^t_i + \theta^*_i + \mu^{t*}v^t_i + \nu^{t*}v^t_i - \mu^{t+1*}(1-\Delta)v^t_i - \nu^{t+1*}(1+\Delta)v^t_i + \sum_{\substack{P \in \mathscr{Q}: \\ i \in P}} \alpha^{kt*}_{iP}, \quad t = 2, \ldots, |T|-1$$

$$\epsilon^{|T|*}_i = p^{|T|}_i + \theta^*_i + \mu^{|T|*}v^{|T|}_i + \nu^{|T|*}v^{|T|}_i + \sum_{\substack{P \in \mathscr{Q}: \\ i \in P}} \alpha^{k|T|*}_{iP} \text{ (with } \theta^*_i \leq 0,\ \mu^{t*} \geq 0,\ \nu^{t*} \leq 0 \text{ and } \alpha^{kt*}_{iP} \leq 0).$$

For both knapsack decompositions, the variables of the subproblem kt will be

$$x_i = \begin{cases} 1 \text{ if stand } i \text{ is selected to belong to region } S \\ 0 \text{ otherwise; } i = k, \ldots, |I|. \end{cases}$$

For the $\mathscr{S}$-knapsack decomposition, the subproblem kt can be formulated by the following integer program:

$$\max \sum_{i=k}^{|I|} \epsilon^{t*}_i x_i + \Omega^{kt*} \tag{20}$$

subject to

$$\sum_{i=k+1}^{|I|} a_i x_i \leq A_{max} - a_k \tag{21}$$

$$x_k = 1 \tag{22}$$

$$x_i \in \{0,1\}; i = 1, \ldots, k. \tag{23}$$

The objective function (20) is to maximize the sum of the node weights ϵ_i^{t*} over the selected nodes and to add the value Ω^{kt*} to the optimal sum. Constraint (22) guarantees that region S contains stand k. Constraints (21) ensure that the area of S does not exceed A_{max}. Constraints (23) state the variable types.

For the $\mathscr{R}$-knapsack decomposition, the pricing subproblem can be formulated as (20), (21), and (23) with additional constraints on the connectivity of the subgraph of $G = (I, E)$ induced by the set $\{i : x_i = 1\}$.

3.2 Knapsack-and-Clique Decompositions

In this section, we define a Dantzig-Wolfe decomposition of the bucket model by reformulating the set defined by constraints (2), (4), (8) and (9). A similar decomposition of the alternative model can also be presented by reformulating the set defined by these constraints and the connectivity constraints.

3.2.1 The Master Problems

Let d and e be the number of variables x_i^{kt} and w_P^{kt} in the bucket model respectively, $e = \sum_{k,t} e(k,t)$, where $e(k,t) = |\{P \in \mathscr{Q} : P \cap \{k, \ldots, |I|\} \neq \emptyset\}|$.

As before, $\mathscr{C}$ can be decomposed into $|I| \times |T|$ sets $\mathscr{C}^{kt} = \{(x,w) \in \{0,1\}^{(|I|-k+1)+e(k,t)} : \sum_{i=k+1}^{|I|} a_i x_i^{kt} \leq (A_{max} - a_k) x_k^{kt}$ and $x_i^{kt} \leq w_P^{kt}; i \geq k; P \in \mathscr{Q} : i \in P\}$.

Again, for a given k, the sets are identical, and an element (x,w) of a set $\mathscr{C}^{kt}$ is either the null vector or a vector $(\bar{x}_S^k, \bar{w}_S^k)$ where S is a region with stands $i \geq k$, containing k and with area not greater than A_{max}, and $\bar{x}_S^k$, $\bar{w}_S^k$ are defined as

$$\bar{x}_{iS}^k = \begin{cases} 1 \text{ if stand } i \text{ belongs to region } S \\ 0 \text{ otherwise} \end{cases}$$

$$\bar{w}_{PS}^k = \begin{cases} 1 \text{ if clique } P \text{ is such that } P \cap S \neq \emptyset \\ 0 \text{ otherwise.} \end{cases}$$

For $t \in T$, $k \in I$ and $S \in \mathscr{S}^k$, we consider again variables y_S^{kt} and y_0^{kt}. The master problem is as follows:

$$\max \sum_{t\in T} \sum_{k\in I} \sum_{S\in\mathscr{S}^k} \sum_{i=k}^{|I|} p_i^t \bar{x}_{iS}^k y_S^{kt} \tag{24}$$

subject to

$$y_0^{kt} + \sum_{S\in\mathscr{S}^k} y_S^{kt} = 1; k \in I; t \in T \tag{25}$$

$$\sum_{k\le\max\{i:i\in P\}} \sum_{S\in\mathscr{S}^k} \bar{w}_{PS}^k y_S^{kt} \le 1; P \in \mathscr{Q}; t \in T \tag{26}$$

$$\sum_{t\in T} \sum_{k=1}^{i} \sum_{S\in\mathscr{S}^k} \bar{x}_{iS}^k y_S^{kt} \le 1; i \in I \tag{27}$$

$$\sum_{i\in I} v_i^t \sum_{k=1}^{i} \sum_{S\in\mathscr{S}^k} \bar{x}_{iS}^k y_{\mathscr{S}}^{kt} \ge (1-\Delta) \sum_{i\in I} v_i^{t-1} \sum_{k=1}^{i} \sum_{S\in\mathscr{S}^k} \bar{x}_{iS}^k y_S^{k,t-1}; t = 2,\ldots,|T| \tag{28}$$

$$\sum_{i\in I} v_i^t \sum_{k=1}^{i} \sum_{S\in\mathscr{S}^k} \bar{x}_{iS}^k y_S^{kt} \le (1+\Delta) \sum_{i\in I} v_i^{t-1} \sum_{k=1}^{i} \sum_{S\in\mathscr{S}^k} \bar{x}_{iS}^k y_S^{k,t-1}; t = 2,\ldots,|T| \tag{29}$$

$$y_S^{kt} \in \{0,1\}; k \in I; t \in T; S \in \mathscr{S}^k \tag{30}$$

$$y_0^{kt} \in \{0,1\}; k \in I; t \in T. \tag{31}$$

We shall refer to this formulation as $\mathscr{S}$-knapsack-and-clique decomposition.

As for the knapsack decomposition, one can consider the $\mathscr{R}$-knapsack-and-clique decomposition corresponding to the reformulation of the bucket model with enforcement of the connectivity of buckets. It turns out that the corresponding master problem coincides with the so-called cluster formulation which has been considered in [5, 8, 9, 11].

The authors in [9] showed that the linear relaxation bound of the cluster formulation remains the same if besides connected clusters, disconnected clusters (buckets) are allowed in the model. This means that in this case the $\mathscr{S}$ and $\mathscr{R}$ knapsack-and-clique decompositions yield the same LP bounds.

The reformulated sets in the $\mathscr{S}$ or $\mathscr{R}$ knapsack-and-clique decompositions are contained in the reformulated sets in the corresponding knapsack decompositions, i.e. they are defined by a tighter set of constraints. This means the LP bounds obtained by the knapsack-and-clique decompositions are never worse than those obtained by the corresponding knapsack decompositions, and it turns out that they

Table 2 Instance for which the LP bound of the knapsack-and-clique decompositions is strictly better than those of the knapsack decompositions

Stand i	a_i	p_i^1	p_i^2	p_i^3	v_i^1	v_i^2	v_i^3	Nodes adjacent to i
1	1.2	121.3	101.9	95.5	165.0	204.8	244.7	2, 3 5
2	0.8	88.5	84.8	62.1	1687.7	1828.3	1940.8	1, 3, 4
3	1.5	11.0	22.7	17.5	1796.5	1946.2	2065.9	1, 2, 4, 5
4	1.0	28.0	64.3	40.2	136.5	204.8	290.2	2, 3, 5, 6, 7
5	0.8	29.1	82.8	48.3	111.6	186.0	297.7	1, 3, 4, 6, 8
6	1.0	124.1	85.1	55.6	125.7	209.5	335.2	4, 5, 7, 8
7	0.5	−21.1	1.9	7.5	859.0	978.9	1098.7	4, 6
8	1.0	47.5	54.7	39.5	32.0	97.0	153.0	5, 6

are better in many instances. Next, we will give an example for which the LP bound of the knapsack-and-clique decomposition is strictly better.

Consider a forest with eight stands (Table 2) and let $A_{max} = 2.3$ and $\Delta = 0.15$. The optimal values of the objective functions of the linear relaxations of the $\mathscr{R}$-knapsack and the $\mathscr{R}$-knapsack-and-clique decompositions are 545.258 and 544.394, respectively. These values are obtained using CPLEX 12.4 [7] as a linear programming solver.

From the above discussion we may state the following results:

Proposition 3 *The LP bound of the $\mathscr{S}$-knapsack-and-clique decomposition is equal to that of the $\mathscr{R}$-knapsack-and-clique decomposition.*

Proposition 4 *The LP bounds of the $\mathscr{S}$ or $\mathscr{R}$ knapsack-and-clique decompositions are better than or equal to those of the $\mathscr{S}$ and $\mathscr{R}$ knapsack decompositions.*

3.2.2 The Pricing Subproblems

For the linear relaxation of the master problem, constraints (30) and (31) are simply replaced by $y_S^{kt} \geq 0$ and $y_0^{kt} \geq 0$ respectively.

Let Ω^{kt}, β_P^t, θ_i, μ^t and ν^t denote the dual variables associated with constraints (25), (26), (27), (28) and (29) of the linear relaxation of the master problem and Ω^{kt*}, β_P^{t*}, θ_i^*, μ^{t*} and ν^{t*} assume an optimal dual solution of the linear relaxation of a restricted master problem. By definition of reduced cost, the objective function of the pricing subproblem kt is given by

$$\max_{S \in \mathscr{S} \mathscr{R}} \{\sum_{i \in S} \epsilon_i^{t*} + \sum_{P: S \cap P \neq \emptyset} \beta_P^{t*}\} + \Omega^{kt*}$$

where $\mathscr{SR} = \mathscr{S}^k$ or $\mathscr{SR} = \mathscr{R}^k$ and

$\epsilon_i^{1*} = p_i^1 + \theta_i^* - \mu^{2*}(1-\Delta)v_i^1 - \nu^{2*}(1+\Delta)v_i^1$
$\epsilon_i^{t*} = p_i^t + \theta_i^* + \mu^{t*}v_i^t + \nu^{t*}v_i^t - \mu^{t+1*}(1-\Delta)v_i^t - \nu^{t+1*}(1+\Delta)v_i^t, t = 2, \ldots, |T|-1$
$\epsilon_i^{|T|*} = p_i^{|T|} + \theta_i^* + \mu^{|T|*}v_i^{|T|} + \nu^{|T|*}v_i^{|T|}$ (with $\theta_i^* \leq 0$, $\mu^{t*} \geq 0$ and $\nu^{t*} \leq 0$).

For both decompositions, the variables of the subproblem kt will be

$$x_i = \begin{cases} 1 \text{ if stand } i \text{ is selected to belong to region } S \\ 0 \text{ otherwise; } i = k, \ldots, |I| \end{cases}$$

$$w_P = \begin{cases} 1 \text{ if at least one stand from clique } P \text{ is selected to belong to region } S \\ 0 \text{ otherwise; } P \in \mathscr{Q} : \max_{i \in P}\{i\} \geq k. \end{cases}$$

For the $\mathscr{S}$-knapsack-and-clique decomposition, the subproblem kt can be formulated by the following integer program:

$$\max \sum_{i=k}^{|I|} \epsilon_i^{t*} x_i + \sum_{P:\max_{i \in P}\{i\} \geq k} \beta_P^{t*} w_P + \Omega^{kt*} \tag{32}$$

subject to

$$x_i \leq w_P; i \geq k; P \in \mathscr{Q} : i \in P \tag{33}$$

$$\sum_{i=k+1}^{|I|} a_i x_i \leq A_{max} - a_k \tag{34}$$

$$x_k = 1 \tag{35}$$

$$x_i \in \{0, 1\}; i = k, \ldots, |I| \tag{36}$$

$$w_P \geq 0; P \in \mathscr{Q} : \max_{i \in P}\{i\} \geq k. \tag{37}$$

The objective function (32) is to maximize the sum of the node weights ϵ_i^{t*} and the clique weights β_P^{t*} over the selected nodes and to add the value Ω^{kt*} to the optimal sum. Constraints (33) ensure that if a node is selected, then any maximal clique with this node is also selected. Constraint (35) guarantees that region S contains stand k. Constraints (34) ensure that the area of region S does not exceed A_{max}. Constraints (36) and (37) state the variable types.

For the $\mathscr{R}$-knapsack-and-clique decomposition, the pricing subproblem can be formulated as (32), (33), (34), (35), (36), and(37) with additional constraints on the connectivity of the subgraph of $G = (I, E)$ induced by the set $\{i : x_i = 1\}$.

4 The Matheuristic

In this section, we present a solution approach for the harvest scheduling problem. This solution approach relies on the $\mathscr{S}$-knapsack-and-clique decomposition, given its advantages over the other decompositions shown in the previous section.

The huge number of decision variables of the decomposition models preclude solving them directly. Therefore, the proposed approach is based on column generation. Given the complexity of the problem, exact methods, as branch-and-price (the combination of CG and branch-and-bound), are not an alternative for large instances. For that reason, the proposed approach is a matheuristic (MatHeur) based on CG and on a general purpose mixed integer programming solver.

The proposed MatHeur is closely related to the general framework "Metaheuristic search by column generation" [1], SearchCol for short. Each iteration of a basic SearchCol algorithm has three main steps: (i) apply CG to the linear relaxation of a perturbed (restricted) master problem, (ii) conduct a search in the space provided by the subproblem solutions obtained so far, and (iii) define perturbations for the next iteration. A perturbation is a constraint that forces a subproblem variable to take value 1 or 0, therefore a perturbed (restricted) master is the original master problem with additional constraints fixing subproblem variables.

Many variants can be devised based on these three steps. The core idea is the exchange of information between CG and *search*. CG provides to *search* its search space and additional information on the subproblem solutions (e.g. their value in the last restricted master problem). *Search* attempts to improve the incumbent which is used, possibly with other solution attributes, to define perturbations to include in CG.

A core concept in SearchCol is that a solution can be represented as being made of subproblem solutions, one from each subproblem. In the $\mathscr{S}$-knapsack-and-clique decomposition, the solutions of subproblem kt are all the regions including stand k and stands with an index higher than k (and also the *empty region*) for period t (variables w are omitted in this discussion for the sake of clarity). With this perspective, a solution to the forestry problem, s, can be represented by $s = (s(1), s(2), \ldots, s(|K|))$, where $s(k)$, $k \in K$, represents the region associated with subproblem k in the solution and K is the set of subproblems.

The main difference between SearchCol and the proposed MatHeur is that in the latter the search is conducted by a general purpose mixed integer programming solver and not by a metaheuristic.

The algorithm of the MatHeur is represented in Fig. 2. Two types of iterations (A and B) are used (almost) alternately. In both, the first step is to apply CG. After CG, a search space is defined.

```
numnoimprov ← 0
typeiter ← A
repeat
    Initialize the set of perturbed subproblems, R, as empty
    Initialize the set of perturbations, P, as empty
    improv ← false
    while R does not include all subroblems do
        Solve CG with the set of perturbations P
        if CG is infeasible then
            Break cycle while
        end if
        Define a search space
        Construct a (partial) solution and provide it to the MIP solver
        Optimize with the MIP solver with a time limit of 0.1 × thenumberofrowsofthedecompositionmodel
        if The incumbent was improved then
            improv = true
        end if
        Define perturbations for 10% of the subproblems (the ones in R are not candidates) and
        include the perturbations in P and the selected subproblems R
    end while
    if improv == true then
        numnoimprov ← 0
        typeiter ← A
    else
        numnoimprov ← numnoimprov + 1
        if typeiter == A then
            typeiter ← B
        end if
        if typeiter == B then
            typeiter ← A
        end if
    end if
until numnoimprov = maxnumnoimprov
```

Fig. 2 The matheuristic. *maxnumnoimprov* is a parameter corresponding to the maximum number of iterations with no improvement

Type A Iteration

The search space is made of the subproblem solutions in the incumbent, all null subproblem solutions, all subproblem solutions generated in the last CG solved, and the $2 * |K|$ subproblem solutions with higher search recency. The search recency of a subproblem solution is the number of times it belonged to global solutions obtained by the mixed integer programming (MIP) solver. The incumbent solution is provided to the MIP solver. After the MIP solver reaches the optimal solution or the time limit imposed, perturbations are defined. The perturbations used in the proposed MatHeur are based on scoring subproblem solutions. The score of a subproblem solution $s(k)$

is obtained through:

$$score_{s(k)} = PresenceInc + LRWeight + CurWeight + CGrecency + SearchRecency + BelongSpace$$

where

- *PresenceInc* is 1 if the subproblem solution is in the incumbent; and 0 otherwise;
- *LRWeight* is the value of the variable of the decomposition model associated with the subproblem solution the first time CG was applied (i.e. in the linear relaxation);
- *CurWeight* is similar to *LRWeight* but considers the last CG solved;
- *CGrecency* is the number of times the subproblem solution was the solution of the subproblem during CG (normalized to a value between 0 and 1);
- *SearchRecency* is the number of times the subproblem solution belonged to global solutions obtained by the MIP solver (normalized);
- *BelongSpace* is the number of times the subproblem solution belonged to search space (normalized).

For each subproblem not yet perturbed, the score of all solutions are calculated and the solution with higher score (excluding the null solution) is selected. Next, the unperturbed subproblems are sorted in descending order of the score of their selected solutions and the first 10 % subproblems are chosen for defining new perturbations. The perturbations consist in forcing all variables with value 1 in the subproblem solution to take value 1 in the next CG by adding constraints to the RMP of CG and take into account their duals in the objective function of the subproblems.

Type B Iteration

There are two differences in a iteration of type B when compared to a iteration of type A. Firstly, the search space is made of all subproblem solutions generated by CG from the start. Secondly, a partial solution is provided to the MIP solver. The partial solution is made of the 20 % subproblem solutions (at most one per subproblem) with higher value of the variable associated with it in the last CG solved.

5 Computational Experiment

5.1 Implementation

The proposed MatHeur was implemented in SearchCol++ (http://searchcol.dps.uminho.pt/), an implementation of the SearchCol framework in C++. When using SearchCol++, only information on the decomposition and problem specific components must be coded. More precisely, the user must provide the number of subproblems, the sense and right hand side of the master constraints, a subproblem solver, how the coefficients in the objective function of the subproblem variables are modified by the duals of the global constraints, and how a column corresponding to a solution of a subproblem is obtained.

The search and perturbations steps of SearchCol are hidden from the user and are controlled through input parameters.

SearchCol++ uses Cplex 12.4 [7] as the linear programming solver for the restricted master problems and also as the MIP solver in the search step. We also used Cplex 12.4 to solve the subproblems. The RMP of CG was initialized with all feasible clusters with two stands at the most. A limit of one hour was set to all approaches testes, including the MatHeur.

Bucket model was solved with the same version of Cplex. The branch-and-cut algorithm was allowed to run for one hour at the most. Computation time includes the time spent by branch-and-bound and the time used to build the model.

In all situations, Cplex default parameters were used throughout, except the ones described in the previous section for the MatHeur. Computations were performed on a desktop computer with an Intel Core i7 – 3.3 GHz processor and with 32 GB RAM.

5.2 Test Instances

We report results for both real and hypothetical test forests (Table 3). Real test forests include Leiria National Forest (LNF) in Portugal and the El Dorado forest in the U.S.A. El Dorado is referred to in [5]. LNF and the hypothetical test forests F and G are referred to in [3]. We also report results for other hypothetical instances which are referred to in [3] (Bloedel and WLC) and [9] (FLG), all partly available at the website www.unbf.ca/fmos/ (El Dorado is in this site as well). The parameter Δ used in the timber flow constraints is 0.1.

We consider two groups of instances: one group with the small and medium instances and the other with the large instances (ElDorado and the three FLG instances).

Table 3 Size of the instances ($\overline{a}_k$ is the average of the stand's area)

Instance	No. nodes	No. edges	No. cliques	$\overline{a}_k$ (ha)	$A_{\max}/\overline{a}_k$	$\lvert T\rvert$
Bloedel	45	112	37	1	4	3
LNF	574	1152	740	14.96	3.46	6
F10x10: −3, −4	100	180	180	1	3, 4	7
F15x15: −3, −4	225	420	420	1	3, 4	7
F20x20: −3, −4	400	760	760	1	3, 4	7
F25x25: −3, −4, −5, −6	625	1200	1200	1	3, 4, 5, 6	7
G15x7: −3, −4	105	247	145	1	3, 4	7
G40x10: −3, −4	400	988	596	1	3, 4	7
G40x14: −3, −4	560	1396	844	1	3, 4	7
G60x10: −3, −4, −5, −6	600	1492	904	1	3, 4, 5, 6	7
WLC	73	98	63	10.12	3.95	7
El Dorado	1363	3609	2041	12.78	3.13	12
FLG-9-1	850	3009	1825	15.22	3.02	11
FLG-10-1	763	2677	1635	15.24	3.02	11
FLG-12-1	2945	10603	6414	15.14	2.97	11

5.3 Results

SearchCol++ also implements branch-and-price (BP) [2] and a heuristic version of it based on the concept of beam search [10] (we name this approach *BeamBP*). We tested those two approaches and a third which consists in solving the RMP obtained at the end of column generation (with no perturbations) but considering that the master variables must be integer (approach named *MipHeur*). For these preliminary tests, we used five instances (F10x10-3, F10x10-4, G15x7-3, G15x7-4, and WLC) and a time limit of one hour. For BP and BeamBP, the variable with a fractional value closest to 0.5 was chosen as the branching variable. For BP, the tree was explored with a dive strategy (when the node generates sons, the up branch one is chose to continue, in the other cases the node with best bound is chosen). In Beam, the width of the beam (i.e. number of nodes selected in each level) was set to three.

We compare MatHeur and MIPHeur with the bucket formulation in Table 4. The quality of the best integer solution found by the bucket model was measured by using the deviation (in percentage) of its value (vis) from the best upper bound found by branch-and-cut (bup): $gap = \dfrac{bup - vis}{vis}100$. The value of gap is returned by Cplex.

The quality of the best integer solutions provided by MatHeur and MIPHeur was measured on the strength of the deviation of their values (vis_{MatH} and vis_{MIPH}) from the best integer solution found by the bucket model, $gap_{\text{MatH/B}} = \dfrac{vis_{\text{MatH}} - vis_B}{vis_B}100$ and $gap_{\text{MIPH/B}} = \frac{vis_{\text{MIPH}} - vis_B}{vis_B}100$, respectively. Therefore, a

Table 4 Computational results for MIPHeur and MatHeur in comparison with the bucket formulation (solved by cplex) for small and medium instances

	Bucket formulation (branch-and-bound)			MIPHeur		MatHeur		
Instance	Best solution	*gap* (%)	Time (sec.)	$gap_{\text{MIPH/B}}$ (%)	Time (sec.)	$gap_{\text{MatH/B}}$ (%)	Time (sec.)	Best approach (best solution)
Bloedel	6676.34	0.008	9	−0.02	3	0	18	Bucket/MatHeur
WLC	7,334,560	0.05	3600	3.4×10^{-5}	239	3.4×10^{-5}	569	MIPHeur/MatHeur
LNF	86,312,900	0.01	196	-1.5×10^{-3}	41	2.8×10^{-3}	2285	MatHeur
F10x10-3	1,960,810	0.52	3600	−1.56	3600	−0.31	3600	Bucket
F10x10-4	1,944,510	1.32	3600	−1.09	3600	−0.25	3600	Bucket
F15x15-3	4,510,440	0.66	3600	−0.15	3600	-2.5×10^{-3}	3600	Bucket
F15x15-4	4,515,580	0.55	3600	−0.08	3600	0.06	3600	MatHeur
F20x20-3	7,951,440	0.50	3604	0.20	3600	−0.06	3600	MIPHeur
F20x20-4	7,981,610	0.43	3612	−0.05	3600	0.18	3600	MatHeur
F25x25-3	12,863,200	0.26	3617	0.15	3600	0.05	3600	MIPHeur
F25x25-4	12,912,100	0.26	3654	−0.04	3600	0.04	3600	MatHeur
F25x25-5	12,918,100	0.21	3725	−0.14	3600	-9.5×10^{-3}	3600	Bucket
F25x25-6	12,887,500	0.45	3843	−0.43	3600	0.19	3600	MatHeur
G15x7-3	2,110,550	1.28	3600	0.08	3600	0.05	3601	MIPHeur
G15x7-4	2,110,330	1.54	3600	−0.77	3600	0.19	3600	MatHeur
G40x10-3	8,110,340	0.27	3606	−0.06	3600	2.9×10^{-3}	3600	MatHeur
G40x10-4	8,096,690	0.46	3619	0.05	3600	-8.4×10^{-4}	3600	MatHeur
G40x14-3	11,300,000	0.24	3618	−0.54	3600	0.06	3601	MatHeur
G40x14-4	11,292,100	0.34	3660	−0.47	3600	0.13	3601	MatHeur
G60x10-3	13,040,400	0.16	3621	−0.16	3600	−0.03	3600	Bucket
G60x10-4	13,036,500	0.19	3672	−0.15	3600	−0.1	3601	Bucket
G60x10-5	13,025,100	0.28	3774	−0.06	3600	0.02	3600	MatHeur
G60x10-6	13,027,800	0.26	3942	−0.09	3600	5.8×10^{-4}	3600	MatHeur

positive value for $gap_{\text{MatH/B}}$ means the MatHeur provided a better solution than the bucket, and the same applies for $gap_{\text{MIPH/B}}$ with respect to the MipHeur and bucket.

For the small and medium instances, MipHeur is clearly inferior to bucket which gave better solutions in 18 instances out of the 23. MatHeur is better than the bucket in 14 instances, worst in 8 and the same solution was obtained in one instance. Comparing the three approaches, MathHeur provided the best solution in 14 instances, bucket in 7 and MipHeur in 4. It is worth noting that for the easiest instance, the Bloedel instance, MipHeur failed to obtain the optimal solution, as the restricted *integer* search space did not contain the necessary subproblem solutions. With the subproblem solutions generated after the perturbations were added, the MatHeur found the optimal solution.

The quality of the best integer solution provided by MatHeur for the small and medium instances was also measured in terms of the deviation of its value from the best integer solution found by MipHeur, $gap_{\text{MatH/MIPH}} = \frac{vis_{\text{MatH}} - vis_{\text{MIPH}}}{vis_{\text{MIPH}}} 100$ (Table 5). MatHeur is better than MIPHeur in 18 instances, worst in 4, and equal in 1.

Table 5 Comparison between MatHeur and MIPHeur in terms of $gap_{\text{MatH/MIPH}}$ for small and medium instances

	$gap_{\text{MatH/MIPH}}$
Bloedel	0.02
WLC	0
LNF	4.2×10^{-3}
F10x10-3	1.28
F10x10-4	0.85
F15x15-3	0.14
F15x15-4	0.15
F20x20-3	−0.26
F20x20-4	0.23
F25x25-3	−0.1
F25x25-4	0.08
F25x25-5	0.13
F25x25-6	0.62
G15x7-3	−0.02
G15x7-4	0.97
G40x10-3	0.06
G40x10-4	−0.05
G40x14-3	0.06
G40x14-4	0.61
G60x10-3	0.13
G60x10-4	0.06
G60x10-5	0.08
G60x10-6	4.3×10^{-3}

Table 6 Computational results for MIPHeur and MatHeur in comparison with the bucket formulation (solved by cplex) for large instances. For the three FLG instances, since the bucket did not provide any solution, no relative quantitative comparison can be made between both approaches. The values displayed between parenthesis are the relative gap of the incumbent and the linear relaxation bound. – means the model was not loaded within one hour. * means a feasible solution was not obtained, the number of violated constraints of the solution with less violated constraints is presented. ** means out of memory

	Bucket formulation (branch-and-bound)			MIPHeur		MatHeur		Best
Instance	Best solution	*gap* (%)	Time (sec.)	$gap_{\text{MIPH/B}}$ (%)	Time (sec.)	$gap_{\text{MatH/B}}$ (%)	Time (sec.)	approach (best solution)
ElDorado	4,210,960	0.06	3902	0.02	3625	1*	3620	MIPHeur
FLG-9-1	–	–	–	(1.60)	3600	4*	3607	MIPHeur
FLG-10-1	–	–	–	(2.76)	3600	**		MIPHeur
FLG-12-1	–	–	–	(0.24)	3600	5*	3636	MIPHeur

The advantage of using a decomposition model becomes clear when analysing the results for the large instances (Table 6). Bucket is not adequate for the three FLG instances since the time spent to construct the model exceeded the time limit imposed. For the ElDorado instance it provided a solution worst than MipHeur. MatHeur could not find feasible solutions in three instances and returned out of memory in the other. As shown in the table, the obtained solutions had a very small number of constraints violated. Providing more time to the search phase of MatHeur would certainly benefit the MatHeur as it would have a behaviour more similar to MipHeur. However, we stuck to the use of the same parameters for all instances. MipHeur is clearly the best approach for the large instances. It provides feasible solutions to the four instances with optimality gaps smaller than 3%.

6 Conclusions

We presented a matheuristic based on column generation for the forest harvest scheduling problems subject to maximum area restrictions. The matheuristic is based on one of the four Dantzig-Wolfe decompositions proposed for the so-called bucket formulation and the bucket formulation with additional constraints on the connectivity of the buckets. The $\mathscr{S}$-knapsack and the $\mathscr{S}$-knapsack-and-clique formulations are decompositions of the bucket model, and the $\mathscr{R}$-knapsack and the $\mathscr{R}$-knapsack-and-clique formulations are decompositions of the bucket model with the connectivity constraints. We proved that the LP bounds of the knapsack-and-clique decompositions are equal, the knapsack-and-clique decompositions dominate the knapsack decompositions, the $\mathscr{R}$-knapsack decomposition dominates the $\mathscr{S}$-knapsack decomposition and the $\mathscr{S}$-knapsack decomposition dominates the bucket model.

We implemented the decomposition model in SearchCol++ allowing testing different solution approaches. Branch-and-price and a heuristic variant where only a subset of nodes in each depth of the tree is considered (we called beam branch-and-price) did not provide feasible solutions in instances where the other tested approaches found solutions with small optimality gaps.

We proposed a matheuristic based on column generation and a general purpose mixed integer programming (MIP) solver. The main idea is the exchange of information from CG to the MIP solver (subproblem solutions, values of the variables in the RMP, among other) allowing defining restricted search spaces and advanced starts in MIP and from the MIP solver to CG (through perturbations based on the incumbent and other measures).

We compared the proposed matheuristic with two approaches. The first one, from the literature, consists in solving a compact model (bucket) with a general purpose solver. The second one consists in applying CG and then solve the integer restricted master problem with a general purpose MIP solver (MipHeur).

The matheuristic was tested with benchmark instances, both real and hypothetical forests, ranging from 45 to 2945 stands, using values of the ratio $A_{\max}/\overline{a}_k$ ranging in the interval [3,7] and temporal horizons from three to twelve periods.

The results show that, for small and medium instances, the proposed matheuristic found better solutions than the bucket formulation and than the MipHeur for the majority of the instances. For large instances (e.g. forests from 763 stands with eleven or twelve periods), the bucket formulation is not effective and the matheuristic failed (by small amounts) to find feasible solutions. The MipHeur was able to obtain solutions with optimality gaps smaller than 3 %.

Acknowledgements This research was partially supported by Fundação para a Ciência e a Tecnologia, projects UID/MAT/04561/2013, PEst-OE/EEI/UI0319/2014 and PTDC/EIAEIA/100645/2008 (SearchCol: Metaheuristic search by column generation). We wish to thank Andres Weintraub and José G. Borges (through the project PTDC/AGR-CFL/64146/2006) for providing some real test forest data.

References

1. Alvelos, F., Sousa, A., Santos, D.: Combining column generation and metaheuristics. In: Talbi, E.G. (ed.) Hybrid Metaheuristics. Studies in Computational Intelligence, vol. 434, pp. 285–334. Springer, Berlin (2013)
2. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: column generation for solving huge integer programs. Oper. Res. **46**, 316–329 (1998)
3. Constantino, M., Martins, I., Borges, J.G.: A new mixed-integer programming model for harvest scheduling subject to maximum area restrictions. Oper. Res. **56**(3), 542–551 (2008)
4. Diestel, R.: Graph Theory. Graduate Texts in Mathematics, 2nd edn. Springer, New York (2000)
5. Goycoolea, M., Murray, A.T., Barahona, F., Epstein, R., Weintraub, A.: Harvest scheduling subject to maximum area restrictions: exploring exact approaches. Oper. Res. **53**(3), 90–500 (2005)

6. Goycoolea, M., Murray, A.T., Vielma, J.P., Weintraub, A.: Evaluating approaches for solving the area restricted model in harvest scheduling. For. Sci. **55**(2), 149–165 (2009)
7. ILOG, ILOG CPLEX 12.4 – User's Manual (2011)
8. Martins, I., Constantino, M., Borges, J.G.: A column generation approach for solving a non-temporal forest harvest model with spatial structure constraints. Eur. J. Oper. Res. **161**(2), 478–498 (2005)
9. Martins, I., Alvelos, F., Constantino, M.: A branch-and-price approach for harvest scheduling subject to maximum area restrictions. Comput. Optim. Appl. **51**, 363–385 (2012)
10. Ow, P.S., Morton, T.E.: Filtered beam search in scheduling. Int. J. Prod. Res. **26**, 35–62 (1988)
11. Vielma, J.P., Murray, A.T., Ryan, D.M., Weintraub, A.: Improving computational capabilities for adressing volume constraints in forest harvest scheduling problems. Eur. J. Oper. Res. **176**(2), 1246–1264 (2007)

A Routing and Waste Collection Case-Study

Karine Martins, Maria Cândida Mourão, and Leonor Santiago Pinto

Abstract Waste collection systems are among the main concerns of municipalities due to the resources involved. In this paper we present a hybrid heuristic to find the vehicle routes that should be performed to collect the household waste along the streets of a network. The solution method hybridizes the resolution of ILP based models with some simple heuristic ideas to assign services (collecting streets) to the vehicles. The Seixal case study, in the Lisbon Metropolitan Area, is tackled and some encouraging results are reported.

1 Introduction

Nowadays, waste collection is a major issue in municipalities as it absorbs an important amount of resources. The routes that vehicles should perform through the streets network with this scope are studied mainly through Vehicle Routing Problems (VRP), whenever the garbage is in dump sites apart, or by Arc Routing Problems (ARP) if it is disposed in small containers along the streets. No matter the last received less attention, it is considered the more adequate in certain situations. That happens in some residential areas of Seixal municipally, the case-study in focus. Recent surveys of waste collection applications may be found in [4], [3] and [5], being the VRP and the ARP problems well studied in [7] and [1].

The county (Concelho) of Seixal is in the Lisbon Metropolitan Area which accommodates about a quarter of the Portuguese population. Spread all over the county there are a number of residential areas of townhouses where the garbage is deposited in small containers along the streets. This option provides a better

K. Martins (✉)
ISEG, ULisboa, Rua do Quelhas, 6, 1200-781 Lisboa, Portugal
e-mail: karineamartins@gmail.com

M.C. Mourão
CIO and ISEG, ULisboa, Rua do Quelhas, 6, 1200-781 Lisboa, Portugal
e-mail: cmourao@iseg.ulisboa.pt

L.S. Pinto
CEMAPRE and ISEG, ULisboa, Rua do Quelhas, 6, 1200-781 Lisboa, Portugal
e-mail: lspinto@iseg.ulisboa.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_15

environment as there is no accumulation of garbage as usually near the big containers. The case-study that we address aims to design these routes and we name the problem as MCARP–Seixal. The MCARP is an ARP defined on a mixed graph, and thus edges are used to represent narrow streets, where the "zigzag" collection is allowed, while arcs stand for one-way streets or large avenues. No matter the capacity requirement is not relevant under this study, each vehicle must perform only one route and a time limit must be observed. The objective is to minimize the total time, i.e. the time traveled by the vehicles to reach and service the streets demanding for refuse collection, and to leave and to return back the depot. In addition, a major concern in the Seixal municipality, included in the so called MCARP–Seixal, is to identify a set of balanced routes, in terms of the traveling times.

We present a model and a heuristic for the MCARP–Seixal. This is a result of an ongoing cooperation with the Seixal municipality that intends to develop software that would consist on a support decision tool for the waste collection routes, incorporating GIS, which, in turn, embodies models and heuristics designed to obtain a set of feasible routes.

The paper is organized as follows. Next section the problem is defined and modeled. Then the heuristic method is detailed. The computational experiments are then reported and analyzed. Finally the conclusions are drawn.

2 Problem Definition and Modeling

The MCARP–Seixal, as mentioned, is an MCARP with a fleet of homogeneous vehicles, with no capacity concerns, but a maximum travel time to observe per vehicle route. Each vehicle must perform only one route that starts and ends at the depot. In parallel to the time minimization objective, MCARP–Seixal also attempts to identify a set of balanced routes, in terms of the traveling times. To model the problem the following notation is needed:

- $\Gamma = (N, A' \cup E)$ is the mixed graph. $A'_R \subseteq A'$ and $E_R \subseteq E$ are the set of arcs and edges demanding for service, respectively, also named as tasks or required links; and N is the set of nodes, representing street crossings, dead-end streets, or the depot.
- $1 \in N$ is the depot node where every vehicle route must start and end. We assume that the depot is far away, with no tasks incident into it, and also coincides with the landfill.
- $G = (N, A)$ is a directed graph where each edge from E is replaced by two opposite arcs, i.e. $A = A' \cup \{(i,j), (j,i) : (i,j) \in E\}$.
- $A_R \subseteq A$ is the set of required arcs in G, ($|A_R| = |A'_R| + 2\,|E_R|$).
- P is the number of routes, equal to the vehicles number.
- L is the maximum time allowed per route (in seconds).
- t^d_{ij} is the deadheading time of arc $(i,j) \in A$ (in seconds), i.e. the time needed to traverse it without serving.

- t_{ij}^s is the service time of arc $(i,j) \in A_R$ (in seconds), i.e. the time needed to collect the refuse along the task.

The scope is to determine a set of P time balanced routes, starting and ending at the depot, observing a time limit L, performing all tasks and minimizing the total time.

2.1 Models

Compact models, i.e. models with a polynomial number of variables and constraints, are derived following [6] and used to find solutions for the problem under study. The next formulation is a valid model for MCARP–Seixal. Let us define:

- $x_{ij}^p = 1$ if $(i,j) \in A_R$ is served by route p, and equal 0 otherwise;
- y_{ij}^p as the number of times that arc $(i,j) \in A$ is deadheaded during route p;
- f_{ij}^p as the flow in arc $(i,j) \in A$, related with the remaining time in route p;
- T as the difference between the total times of the two most different routes (in seconds), defined for the routes balancing objective, and thus referred as the balancing variable.

$$\text{(FB1)} \quad \min \sum_{p=1}^{P} \left(\sum_{(i,j)\in A} t_{ij}^d y_{ij}^p + \sum_{(i,j)\in A_R} t_{ij}^s x_{ij}^p \right) + T \tag{1}$$

Subject to:

$$\sum_{j:(i,j)\in A} y_{ij}^p + \sum_{j:(i,j)\in R} x_{ij}^p = \sum_{j:(j,i)\in A} y_{ji}^p + \sum_{j:(j,i)\in R} x_{ji}^p, \quad \forall i \in N,\ \forall p \tag{2}$$

$$\sum_{p=1}^{P} x_{ij}^p = 1, \quad \forall (i,j) \in A'_R \tag{3}$$

$$\sum_{p=1}^{P} \left(x_{ij}^p + x_{ji}^p \right) = 1, \quad \forall (i,j) \in E_R \tag{4}$$

$$\sum_{j:(1,j)\in A} y_{1j}^p = 1, \quad \forall p \tag{5}$$

$$\sum_{j:(j,i)\in A} f_{ji}^p - \sum_{j:(i,j)\in A} f_{ij}^p = \sum_{j:(j,i)\in A_R} t_{ji}^s x_{ji}^p + \sum_{j:(j,i)\in A} t_{ji}^d y_{ji}^p, \quad \forall i \in N,\ \forall p \tag{6}$$

$$\sum_{j:(1,j)\in A} f_{1j}^p = \sum_{(i,j)\in A_R} t_{ij}^s x_{ij}^p + \sum_{(i,j)\in A} t_{ij}^d y_{ij}^p, \quad \forall p \tag{7}$$

$$\sum_{i:(i,1)\in A} f_{i1}^{p} = \sum_{i:(i,1)\in A_R} t_{i1}^{s} x_{i1}^{p} + \sum_{i:(i,1)\in A} t_{i1}^{d} y_{i1}^{p}, \qquad \forall p \tag{8}$$

$$\sum_{j:(1,j)\in A} f_{1j}^{p} \geq \sum_{j:(1,j)\in A} f_{1j}^{p+1}, \qquad p = 1, \ldots, P-1 \tag{9}$$

$$\sum_{j:(1,j)\in A} f_{1j}^{1} - \sum_{j:(1,j)\in A} f_{1j}^{P} \leq T \tag{10}$$

$$f_{ij}^{p} \leq L(x_{ij}^{p} + y_{ij}^{p}), \qquad \forall (i,j) \in A,\ \forall p \tag{11}$$

$$x_{ij}^{p} \in \{0.1\}, \qquad \forall (i,j) \in A_R,\ \forall p \tag{12}$$

$$f_{ij}^{p} \geq 0, \qquad \forall (i,j) \in A,\ \forall p \tag{13}$$

$$y_{ij}^{p} \geq 0 \quad \text{integer}, \qquad \forall (i,j) \in A,\ \forall p \tag{14}$$

The objective function (1) represents the total time plus the balancing variable T. This variable is defined to balance the routes, i.e. to try to find routes with similar traveling times. Constraints (2) state that the number of times a vehicle enters and leaves a vertex is equal. Constraints (3) and (4) impose that tasks are served only once, (5) require that each vehicle performs one route starting at the depot, and (6), (7), and (8) are needed to ensure the connectivity of the routes, which jointly with (2) also guarantee its continuity. Constraints (9) and (10) establish the maximum time between two routes, and define T variable. Constraints (11) relate the flow variables with the remaining variables, also needed to impose the time limit constraint per route.

As in the case-study of Seixal, the above formulation assumes that the depot is far away from the demand network, with no tasks incident into it. Then, the depot will never be used as an intermediate node in a route.

To try to reduce the computation time needed to solve this ILP we add lower bounds on the values for the flow variables, that is:

$$f_{ij}^{p} \geq (t_{ij}^{s} + minT) x_{ij}^{p} \quad \forall (i,j) \in A_R,\ \forall p \tag{15}$$

$$f_{ij}^{p} \geq (t_{ij}^{d} + minT) y_{ij}^{p} \quad \forall (i,j) \in A,\ \forall p \tag{16}$$

where $minT = \min_{k\in\{d,s\};(a,b)\in A}(t_{a,b}^{k})$.

This model differs from the model in [6] by considering: (i) flow variables related with the time instead of service and thus modifying (6), (7), and (8); (ii) a new variable, T and a new constraint (10); and (iii) new flow lower bound constraints (15) and (16).

Other models are also considered to try to infer about the effect of introducing the balancing requirements this way, named as (F) and (FB2). (F) does not consider route balancing at all, and thus T variable is dropped and so constraints (10), as in an MCARP model. Note that (9) are kept only for breaking symmetries. In model (FB2) we use T as a parameter, and so it is only removed from the objective function,

the set of feasible solutions is also defined by (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), and (16).

Another model that is helpful is the aggregated model, (Ag), similar to the one in [6]. In this model, variables are aggregated over the set of routes yielding:

- $x_{ij} = 1$ if $(i,j) \in A_R$ is served, and equal 0 otherwise; thus $x_{ij} = \sum_{p=1}^{P} x_{ij}^p$;
- y_{ij} the number of times that $(i,j) \in A$ is deadheaded, thus $y_{ij} = \sum_{p=1}^{P} y_{ij}^p$;
- f_{ij} the flow that traverses $(i,j) \in A$, thus $f_{ij} = \sum_{p=1}^{P} f_{ij}^p$.

Then, (Ag) may be written from any of the first three models (FB1, FB2 or F) by dropping T variable, by summing over p expressions (1), (2), (5), (6), (7), (8), (11), (12), (13), and (14), and by replacing the variables as above defined.

3 Heuristic

As integer solvers usually fails to find solutions for bigger instances, we also propose a heuristic method. This heuristic mixes the resolution of both aggregate and valid models with some simple heuristic rules to assign tasks to vehicles reducing this way the instances dimensions.

In the heuristic it is assumed that all nodes of graph G (with the exception of the depot) have tasks incident into it. There is no loss of generality, as a deadheading path between two end task nodes may be replaced by an arc with an associated length (time or cost) given by the shortest path length (time or cost) between the two corresponding end task nodes.

The heuristic is structured in three phases.

Heuristic

1. Solve the aggregate model (Ag)
2. Fix tasks
   (a) Compute time distances
   (b) Select seeds
   (c) Assign tasks to vehicles
      A priori assignment
      A posteriori assignment
3. Solve the valid model considering the already fixed tasks (Phase 2). This model is named as FH.

The resolution of the aggregate model in Phase 1 produces a giant route. The value of its solution is a lower bound on the optimal value of the problem. The graph

induced by the solution of the aggregate model, $G_I = (V, A_I)$ is a subgraph of G, $A_I \subset A$. Usually, it includes only a subset of the deadheading links and identifies the edge tasks service direction by choosing only one of the two opposite arcs associated with each edge task to serve.

Phase 2 of the heuristic proceeds with graph G_I, and is the main focus of this work. This aims to assign some tasks to vehicles to reduce the dimension of the problem in Phase 3, that consists in solving the valid model with the tasks prefixed. In summary, in Phase 2 a special vertex, seed-node, is selected for each route and from it, it is identified a circuit starting and ending at the depot, and that passes through the seed, including a maximum percentage, α (in the tests we set $\alpha = 0.8$), of the average demand per route, $\overline{Q}$. Although in this phase all the links in the circuit are needed, only the information about the tasks fixed is conveyed to Phase 3. Phase 2 is next detailed.

3.1 Compute Time Distances in G_I

The time distance between any pair of nodes $i, j \in V$, $Dist(i, j)$, is the duration of the corresponding shortest path in G_I, assuming that the time of arc $(u, v) \in A_I$ is given by t^d_{uv}. On that purpose the Floyd algorithm (see [2]) was used. Based on these time distances it is also computed:

- $DCirc_{1j} = Dist(1, j) + Dist(j, 1), j \in V \setminus \{1\}$, the duration of the shortest circuit that includes vertex j and the depot, referred by $Circ(1, j)$;
- $\overline{Dist} = \frac{\sum_{i,j \in V} Dist(i,j)}{N^2 - N}$ the average time distance between all nodes.

3.2 Seeds Selection

To fix tasks that may promote balanced routes we start to find a set of seed–nodes spread all over the graph. A set of P nodes (one per route), far away from each other and from the depot, within a given time from the depot, to try to ensure that each vehicle can return to the depot within the time limit, is thus selected. The distances between the seeds and the depot will be controlled by parameter D_A, and the distances between two seed-nodes through D_B. Initially, we set $D_A = D_B = \overline{Dist}$. In the absence of seed-node candidates, D_A and D_B are successively divided by a factor, $\theta > 1$ (in the tests $\theta = 1.1$), until the desired number of seed-nodes is met. Beyond these controls, and to try to ensure that from each seed-node a feasible tour may be found, the time of the minimum circuit including the seed and depot may not exceed a given percentage, ω (in the tests $\omega = 0.6$), of the maximum time route, L. The first seed to be selected, $k \in V \setminus \{1\}$, is thus a task-node away from the depot at least D_A, and which maximizes $DCirc_{1j}$ amongst the ones for which $DCirc_{1j} \leq \omega L$. That is, defining Vs as the set of seeds (initially empty), and

$\Omega = \{j \in V \setminus (Vs \cup \{1\}) : Dist(1,j) \geq D_A \wedge Dist(j,1) \geq D_A \wedge DCirc_{1j} \leq \omega L\}$, the first seed-node is $seed^1 = arg(\max_{j\in\Omega}(DCirc_{1j}))$. Each following seed is selected among the nodes away from the depot and from the previous selected seeds at least D_B, i.e. each new seed is given by $seed^p = arg(\max_{j\in K}(Circ_{1j}))$, where $K = \{j \in \Omega : Dist(i,j) \geq D_B \wedge Dist(j,i) \geq D_B, \forall i \in Vs\}$.

3.3 Assign Tasks to Vehicles

Two types of assignments are considered: (i) a priori and (ii) a posteriori.

A Priori Assignment

This assignment aims to find two paths, one from the depot to a seed node and the other from the seed back to the depot, and to fix tasks in both paths so that it could be a wise skeleton of a route. These paths, although allowing an initial deviation from the depot, not greater than Dev (detailed later in this section), are built to prevent that a route deviates uncontrollably from it.

Four rules (R0, R1, R2 or R3) are applied to assign links to a route. Rule R0 must be satisfied by all candidate tasks, being then used jointly and sequentially with one of the rules R1–R3.

On this purpose, two specific nodes linked with the seed-node through a directed path, DP, in G_I are considered: (i) node v_i, the initial node of $DP(v_i, seed)$; and (ii) node v_j, the end-node of $DP(seed, v_j)$, as is shown in Fig. 1.

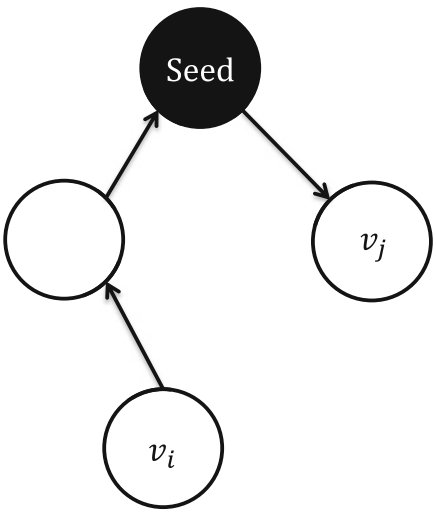


Fig. 1 Nodes linked with the seed

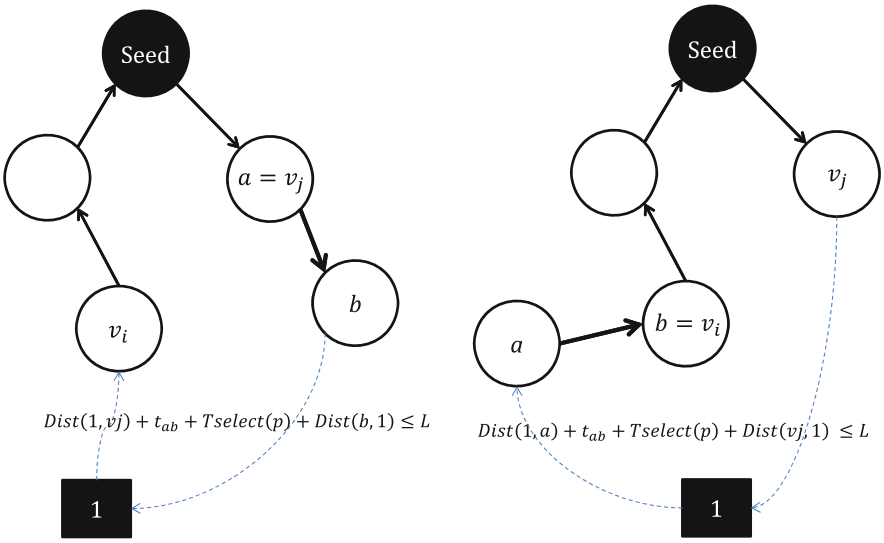


Fig. 2 Candidate links accordingly rule R0

The method, starts with both v_i and v_j equal to the seed-node, and iteratively moves these nodes away from the seed through the identified directed paths in G_I. The tasks to be fixed to a route (identified by its seed-node) belong to these directed paths, which in turn define a directed path from v_i to v_j, throughout the seed, $DP(v_i, seed, v_j)$, and are chosen accordingly to the following rules.

Rule R0 – within this rule the candidate links try to ensure that a feasible route may be found, i.e. the time needed to service the assigned links from the depot is compatible with the time limit L. Let $TSelect(p)$ be the total time of the path $DP(v_i, seed^p, v_j)$ for route p and $TaskFixed$ be the set of already fixed tasks, the R0-candidate links, as illustrated by thicker arcs in Fig. 2, are $(a, b) \in A : a = v_j \vee b = v_i$ and

$$\begin{cases} Dist(1, v_i) + t_{ab} + Tselect(p) + Dist(b, 1) \leq L, \text{ if } a = v_j \\ Dist(1, a) + t_{ab} + Tselect(p) + Dist(v_j, 1) \leq L, \text{ if } b = v_i, \end{cases} \tag{17}$$

where, t_{ab} is the service time in case (a, b) is tackled as a task, or the deadheading time, otherwise.

Then, as referred, R0 is used together with the sequence R1, R2 and R3. In order to explain it, assume that $Qfix(p)$ is the total demand in path $DP(v_i, seed^p, v_j)$ assigned to route p.

Rule R1 – looks for a task that do not fill the route more than a percentage of the average demand and although allowing a deviation from the depot prevents that it exceeds a given parameter, named by Dev. Then, R1–candidate tasks, as illustrated

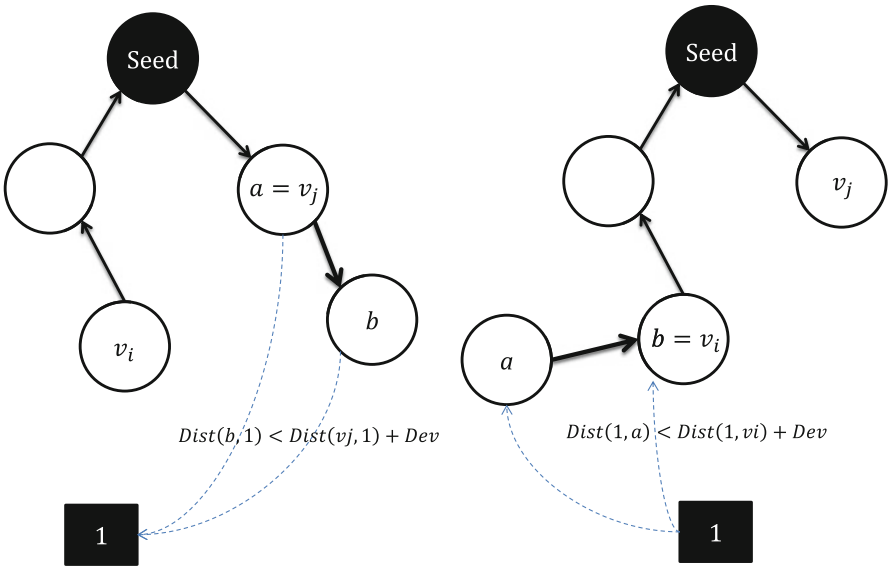


Fig. 3 Candidate tasks accordingly rule R1

by thicker arcs in Fig. 3, are $(a,b) \in A_R \setminus TaskFixed : (a = v_j \vee b = v_i)$ and $q_{ab} + Qfix(p) \leq \alpha\overline{Q}$ and

$$\begin{cases} Dist(b,1) < Dist(a,1) + Dev, \text{ if } a = v_j \\ Dist(1,a) < Dist(1,b) + Dev, \text{ if } b = v_i \end{cases} \tag{18}$$

If no tasks verify R1, we study the rule R2, which allow the selection of tasks that deviate the route from the depot more than Dev.

Rule R2 – aims to find a task, which although being farther from the depot, than nodes v_i or v_j, more than Dev, belongs to a sub-circuit in graph G_I with a small number of links, less than a specified parameter, t (in the tests $t = 3$). The rational is to allow a deviation from the depot through a task that being in a circuit with one of the specific nodes (v_i or v_j) ensures a short way back. Let $Nlink(b,a)$ be the number of links in the sub-circuit, $Circ(b,a)$, linked with the specific node $a = v_j \vee b = v_i$ excluding task (a,b) (see Fig. 4). Then, $R2$–candidate links are $(a,b) \in A_R \setminus TaskFixed : (a = v_j \vee b = v_i)$ and

- $q_{ab} + Qfix(p) \leq \alpha\overline{Q}$ and
- $Nlink(b,a) < t$

In the absence of tasks satisfying R2 some deadheading links may be considered following rule R3.

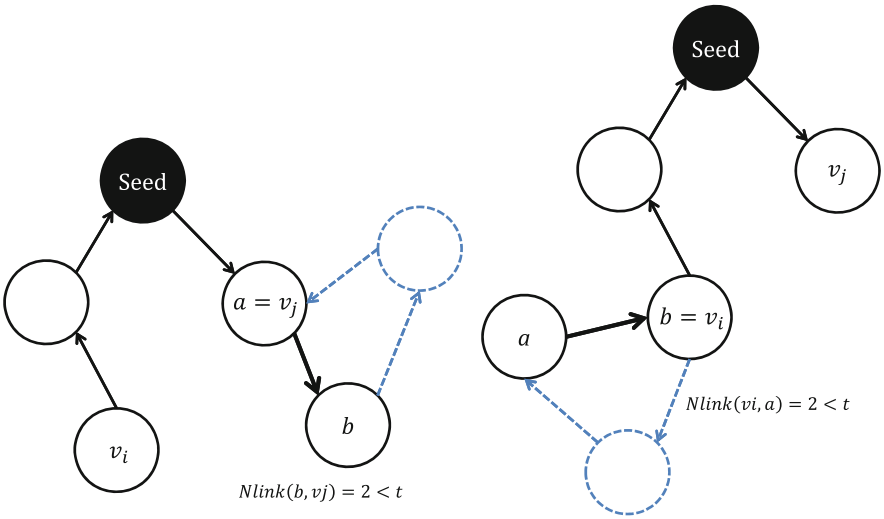


Fig. 4 Candidate tasks accordingly rule R2, with $t = 3$

Rule R3 – this rule, although similar to R1, looks for a link (instead of a task) which does not deviate from the depot more than Dev. R3–candidate links are $(a, b) \in A : (a = v_j \vee b = v_i)$ and

$$\begin{cases} Dist(b, 1) < Dist(a, 1) + Dev, \text{ if } a = v_j \\ Dist(1, a) < Dist(1, b) + Dev, \text{ if } b = v_i \end{cases} \tag{19}$$

For each rule, and in case of more than one candidate, we select the one that maximizes the distance between the designed sub-route and the depot. For Phase 3 of the heuristic only the fixed tasks are relevant, the identified deadheading links are just needed for the a posteriori assignment (detailed in the next section).

The above mentioned deviation, Dev, is made positive while $Qfix(p) \leq \mu\overline{Q}, (0 < \mu < \alpha)$ and null thereafter. Thus, it is allowed an initial deviation from the depot. To keep this parameter within meaningful values we compute $Dev = \frac{DMinSeed}{r}$ (we set $r = 4$), where $DMinSeed$ is the minimum distance between all pairs of seed–nodes.

A priori assignment is repeated until a circuit per vehicle between the depot and its seed-node is identified in two distinct ways: (i) Route to Route (RR) or (ii) Multi-Route (MR). Route to Route starts with a seed (route) and repeats the assignment process as long as possible in the same route (i.e. until the depot is reached or one of the percentages settled bounds for time or demand is achieved), after that chooses another seed (route), and so on. Accordingly Multi-Route procedure the assignment is sequentially done to all the routes. Firstly, one task per route is assigned, secondly, a second task per route is identified, until reaching the depot throughout the directed paths, the time or the demand assigned percentage. In both cases, RR or MR, the sequence of the routes is given by the order of the seeds (1st route – 1st seed selected,

2nd route – 2nd seed selected, etc.). With the MR strategy one tries to even the routes, avoiding the worst choices to be left for the last routes.

In fact, we will not present the results for (RR) strategy, as our preliminary tests confirm that (MR) clearly overcomes (RR).

A Posteriori Assignment

With the a priori assignment we notice that usually the paths $DP(v_i, seed, v_j)$ reach the depot, not allowing further assignments, with only a small percentage of the demand fixed. This motivated the development of the a posteriori assignment, which assigns the tasks to the closest route, within the lime limit. A process similar to Clarke–Wright's savings method (see [2]) is used. An unassigned task is randomly selected, and the cost to insert task (a, b) in route k before link (u, v) (see Fig. 5) is given by:

$$Cost(a,b,u,k) = \begin{cases} Dist(u,a) + Dist(b,u), & \text{if } (u,v) \text{ is served} \\ Dist(u,a) + Dist(b,v) - t^d_{u,v}, & \text{if } (u,v) \text{ is deadheaded} \end{cases} \tag{20}$$

For each task (a, b) the best position and the best route for its insertion is evaluated by computing:

- $Cost(a,b,k) = \min_{u:(u,v)\in Circ(1,seed^k)} \{Cost(a,b,u,k)\}$ and
- $Cost(a,b) = \min_k \{Cost(a,b,k)\}$.

Let u^k the best insertion point for the unassigned task (a, b), in route k, i.e. $u^k = \arg(\min_{u:(u,v)\in Circ(1,seed^k)} \{Cost(a,b,u,k)\})$, and p be the best route to insert (a, b), i.e. $p = \arg(\min_k \{Cost(a,b,k)\})$.

The a posteriori assignment iteratively, and if possible, assigns tasks to routes as summarized in the next procedure.

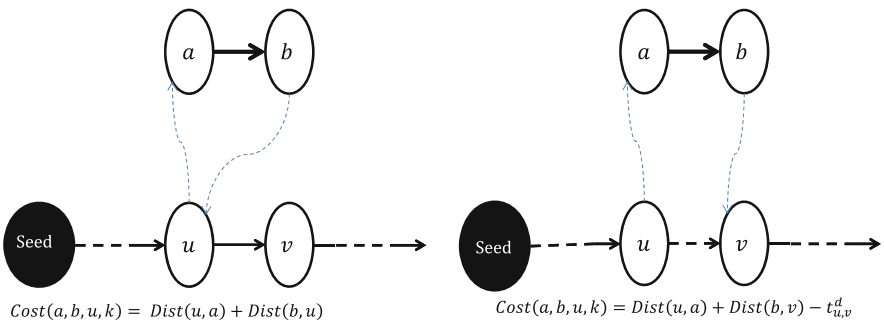


Fig. 5 Best insertion position for a posteriori assignment

A posteriori assignment procedure
Let $TF \leftarrow A_R \setminus TaskFixed$
Repeat

1. Randomly select $(a, b) \in TF$
2. Let $CR \leftarrow \emptyset$ be the set of candidate routes to insert (a, b)
3. For each route $k = 1, \ldots, P$, identify u^k the best insertion position for (a, b)

   If $(Cost(a, b, k) + t^s_{ab} + Tselect(k) \leq L) \wedge (q_{u^k v} + Qfix(k) \leq \alpha \overline{Q})$
   Then update $CR \leftarrow CR \cup \{k\}$

4. **If** $(CR \neq \emptyset)$

   Then Identify route $p = \arg(\min_{(k \in CR)} \{Cost(a, b, k)\})$
   Update $TaskFixed \leftarrow TaskFixed \cup \{(a, b)\}$
   Add (a, b) to route p before/after node u^p

5. $TF \leftarrow TF \setminus \{(a, b)\}$

Until $(TF = \emptyset)$

4 Computational Results

The results were obtained with CPLEX 12.3, on a machine with Intel(R) Pentium(R) CPU B950, 2,1 GHz (6 RAM). Instances from the case study of Seixal were generated and used to assess the performance of the heuristic. Next the instances are then first characterized, followed by the presentation and discussion of the obtained computational results.

Computational results aim to: (i) evaluate the performance of the heuristic; (ii) evaluate the impact of the balancing impositions both in the model and in the quality of the solutions for their practical implementation.

4.1 Instances

Real world based instances are generated to test the proposed heuristic. Seixal historical data regarding the waste collection system was treated and used to compute the links parameters, as the amount of refuse per task, the service and deadheading times. Seixal map is divided into several zones, accordingly the routes in use and devised by the Waste Division. The data set characteristics are displayed in Table 1. The instances dimensions, vary between 106 and 257 nodes and from 143 to 439 links.

Table 1 Characteristics of the Seixal instances

Instance	$\|V\|$	$\|A \cup E\|$	$\|A_R\|$	$\|E_R\|$	P	$\overline{Q}$
S1	106	143	10	87	2	250
S2	148	284	63	119	3	6042
S3	167	320	77	70	2	2571
S4	179	352	43	120	2	4324
S5	257	439	47	235	3	5043

4.2 Analysis of Computational Results

Table 2 presents gap values for the different alternatives under study. The two columns after the instance names depict the gap values for the aggregate relaxation (Ag) and for the MCARP model (F), measuring the distance between the respective upper bound and the MCARP lower bound. The remaining columns always compare two upper bounds, computed in two different ways. Columns 4–9 compare upper bounds of model (FH, FB2 or FB1) against MCARP upper bound values, in case the heuristic is performed with parameter $\mu = 0$ (columns 4–6), not allowing further deviations from the depot, or with a deviation of 10 % (columns 7–9). The last three columns measure the effect of the μ parameter on the solution values, comparing the respective upper bounds.

Recall that (FB1) is an extended MCARP model, where the balancing impositions on the time limit are considered through as extra variable T, whereas (FB2) uses T as a parameter, fixed after the resolution of model (F). As may be seen, the valid MCARP model (F) is able to solve all the instances but S2. While no cpu time limit was imposed for model (F), an one hour limit was settled to run the remaining ILP models.

When the balancing is imposed the quality of the solution values increase considerably, as may be confirmed through columns 4–9, and none of the models FB1 or FB2 seems to be consistently better. However, by Table 3 FB1 is systematically more effective than FB2 in achieving a good balance. In fact, Table 3 depicts the imbalance of the solutions, computed as: $Imb = \frac{Tmax - Tmin}{Tmin} \times 100$, where $Tmax$ and $Tmin$ are, respectively, the time spent on the more consuming and the less consuming route. Note that the increasing on parameter μ, from 0 to 10, produced clearly more equilibrium in the routes designed.

We stress that the time consuming part of the heuristic is, as expected, spent on the integer solver that usually does not end up with an optimal solution. In fact, Phase 2 of the heuristics takes only a few seconds, whilst Phase 1 and 3 usually attain the imposed time limit.

Table 2 Gap values

	No Heuristic		Aggregated model + Heuristic + Final model								
			(U#−UF)/UF						(U#10−U#0)/U#0		
	(U−Lb)/Lb		μ=0			μ=10					
Instance	Ag(F)	F	FH	FB2	FB1	FH	FB2	FB1	FH	FB2	FB1
S1	−0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S2	0.03	0.09	0.00	43.42	3.65	0.00	6.31	2.27	0.00	7.54	−1.33
S3	0.00	0.00	0.00	2.57	2.42	0.00	2.46	2.44	0.00	−0.10	0.02
S4	0.00	0.00	0.00	5.67	5.67	0.00	5.67	5.50	0.00	0.00	−0.16
S5	0.52	0.00	3.34	12.17	22.52	3.82	11.67	21.25	0.46	0.01	−0.58

Legend: Lb – lower bound for the MCARP model (F); Ag(F) – aggregated MCARP; UF is (F) upper bound; (FH) – (F) with prefixed tasks; (FB1) – (FH) with balancing constraints (9)-(10); (FB2) – (FB1) with variable T in the objective; $U\#$ is the upper bound for model # (FH, FB1 or FB2); $U\#\mu$ – $U\#$ with fixed μ

Table 3 Imbalance (in %)

Imb		$\mu = 0$			$\mu = 10$		
Instance	F	FH	FB2	FB1	FH	FB2	FB1
S1	3.48	3.48	3.48	0.00	0.00	0.00	0.00
S2	31.63	31.63	31.63	0.00	0.00	0.05	0.04
S3	39.30	39.30	39.30	0.00	0.00	0.00	0.00
S4	12.03	12.03	12.03	0.00	0.00	0.03	0.03
S5	2.71	2.97	11.43	0.00	0.00	0.39	0.18

Legend: (F) – MCARP model; (FH) – (F) with prefixed tasks; (FB1) – (FH) with balancing constraints (9)-(10); (FB2) – (FB1) with T in the objective function

5 Conclusions

In this work the MCARP–Seixal is defined to approach a waste collection problem in a county in the metropolitan area of Lisbon. A compact integer formulation is derived and a relaxation is presented as well. Those models are hybridized with a heuristic procedure to produce feasible solutions. The method is tested on real instances and is quite promising in providing solutions with balance sets of routes.

This works is part of an ongoing project, that will proceed with its integration in a Decision Support System that will include an automatic data gathering with the help of a GIS, that will also assist in a friendly solutions presentation.

Acknowledgements Thanks are due to João Janela and Mafalda Afonso for the work with the Seixal data gathering. Authors wish to thank National Funding from FCT (PTDC/ECE-GES/121406; PEsT-OE/EGE/UI0491; PEsT-OE/MAT/UI0152) for support.

References

1. Dror, M.: Arc Routing: Theory, Solutions and Applications. Kluwer, Boston (2000)
2. Evans, J.R., Minieka, E.: Optimization Algorithms for Networks and Graphs. Marcel Dekker, New York (1992)
3. Ghiani, G., Mourão, C., Pinto, L., Vigo, D.: Routing in waste collection applications. In: Corberán, Á., Laporte, G. (eds.) Arc Routing Problems, Methods, and Applications, chapter 15. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (2014)
4. Ghiani, G., Laganà, D., Manni, E., Musmanno, R., Vigo, D.: Operations research in solid waste management: a survey of strategic and tactical issues. Comput. Oper. Res. **44**, 22–32 (2014)
5. Golden, B.L., Assad, A.A., Wasil E.A.: Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Toth P., Vigo, D. (eds.) The Vehicle Routing Problem, pp. 245–286. SIAM, Philadelphia (2002)
6. Gouveia, L., Mourão, M.C., Pinto, L.S.: Lower bounds for the mixed capacitated arc routing problem. Comput. Oper. Res. **37**, 692–699 (2010)
7. Toth, P., Vigo, D. (eds.): The Vehicle Routing Problem. SIAM, Philadelphia (2002)

Exact Solutions to the Short Sea Shipping Distribution Problem

Ana Moura and Jorge Oliveira

Abstract Short sea shipping has several advantages over other means of transportation, recognized by EU members. The maritime transportation could be dealt like a combination of two well-known problems: the container stowage problem and routing planning problem. The integration of these two well-known problems results in a new problem CSSRP (Container stowage and ship routing problem) that is also an hard combinatorial optimization problem. The aim of this work is to solve the CSSRP using a mixed integer programming model. It is proved that regardless the complexity of this problem, optimal solutions could be achieved in a reduced computational time. For testing the mathematical model some problems based on real data were generated and a sensibility analysis was performed.

1 Introduction

There are several transportation systems that can be used to transport containers from one destination to another. Transport over sea is among the various forms of transportation and the one with a greatest growth rate over the last decades. Presently, short sea shipping is responsible for a significant part of all freight moved within the European Union borders. In the last decades containerization has revolutionized cargo shipping. According to Drewry Shipping Consultants (http://www.drewry.co.uk/), today over 70 % of the value of the world international seaborne trade is being moved in containers. The world container fleet expanded at an average annual growth rate of 9 %. A significant part of all freight moved within the European Union travels by sea. EU statistics state that 3800 million tons

A. Moura (✉)
Department of Economics, Management and Industrial Engineering, CIDMA – Center for Research and Development in Mathematics and Applications, University of Aveiro Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
e-mail: ana.moura@ua.pt

J. Oliveira
Department of Economics, Management and Industrial Engineering, University of Aveiro, Portugal, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
e-mail: jorge.antonio@ua.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_16

were transported by ship in 2006 and it is hoped that, in 2018, if the economic crisis overcome, the 5300 million tons mark will be reached. Short sea shipping has several advantages but also has some downsides: the bureaucracy attached to customs and ports; port services costs and efficiency; travel duration; inflexibility of routes; and dependency on environmental factors. Considering some of these downsides, this work intends to minimize the transportation costs reducing the travel duration and the port servicing cost and efficiency. With that purpose, a mixed integer programing (MIP) model that minimizes the total cost distribution was developed.

The motivation and the idea for this work arise from the parallelism between this type of distribution and the distribution problems over land. In this latter type of transportation a well-known problem, the Vehicle Routing and Loading Problem (VRLP) that is an integration of the Vehicle Routing Problem (VRP) and Container Loading Problem (CLP), was first presented by [15]. The routes and the packing of boxes (or pallets) in the trucks are computed considering the Last-In-First-Out (LIFO) strategy and all the constraints related to this two different problems. The study of VRLP only started in 2006 and since then further work has been done. Gendreau et al. [13] and Moura and Oliveira [15] were the first authors to integrate the VRP and CLP and to publish benchmark instances. Moura and Oliveira [15] considered a mathematical formulation to the VRLP with Time Windows and tri-dimensional packing and in this work they presented two heuristic approaches, a hierarchical and a sequential approach. Bearing this in mind, we applied the same concept to short sea distance distribution problems. Nevertheless, the main difference between these two apparently similar problems is in the cargo loading. The routes are more dependent on the loading plans, due to both the vessel and cargo stability and the great amount of time that each vessel spends in a port in order to load and unload the containers. A good loading plan in this case is more crucial to the entire distribution process. The routes are planed depending on: distances, demands and the delivery deadlines of each port. Besides, the mathematical model considers the interconnection between the routes optimization and the containers loading on containerships, in order to reduce overstowing.[1] According to these, two different kinds of optimization problems must be dealt with, in an integrated manner: the Containership Routing Problem with Deadlines and the Container Stowage Problem. The integration of these two problems was named by [16] as the Container Stowage and Ship Routing Problem (CSSRP).

This paper is structured as follows: Sect. 2 consists of a detailed description of the CSSRP; Sect. 3 is a review of the relevant literature; in Sect. 4 a mixed integer programming model of CSSRP is presented; Sect. 5 is dedicated to the discussion of results, considering some generated problem instances, and the analysis of the model performance; to conclude in Sect. 6, some global remarks are presented.

[1] An overstow occurs when there are containers that must be moved because they block the access to other containers that have to be unloaded.

2 Container Stowage and Ship Routing Problem Description

The containerships are vessels (for now on always just referred to as vessels) specifically constructed to transport containers. The cargo space of the vessel is divided in bays positioned on deck and below the deck of the ship and separated by a hatch cover [18]. Each bay (longitudinal sections) consists of a set of stacks (transversal sections) and tiers (vertical sections of a stack) that are one container wide (Fig. 1). The space related to one stack and tier is usually referred by a slot, which is a single space where containers could be loaded. The vessels have capacity constraints, related to weight, slot dimensions and number of slots. Although there are different types of containers with different dimensions and characteristics, in this work only the containers types considered in our problem are described. Each container has a weight, a destination port and standard dimensions. Those standard dimensions are measured in foot equivalent units and could be TEU (Twenty-foot Equivalent Unit) or FEU (Forty-foot Equivalent Unit). In each slot one 40′ or two 20′ can be loaded. Another type of containers are those that need electric power (refrigerated containers). These particular containers must be loaded in slots that have a power plug. The maximum weight of each container is 24 tons for 20′ and 40 tons for 40′.

In CSSRP we have an origin port with a set of vessels and containers. The vessels must deliver the containers to other different ports within a given period of time (deadlines), each port has a demand (set of containers) that must be satisfied. In the end of the distribution process, the vessels must return to the port of origin with

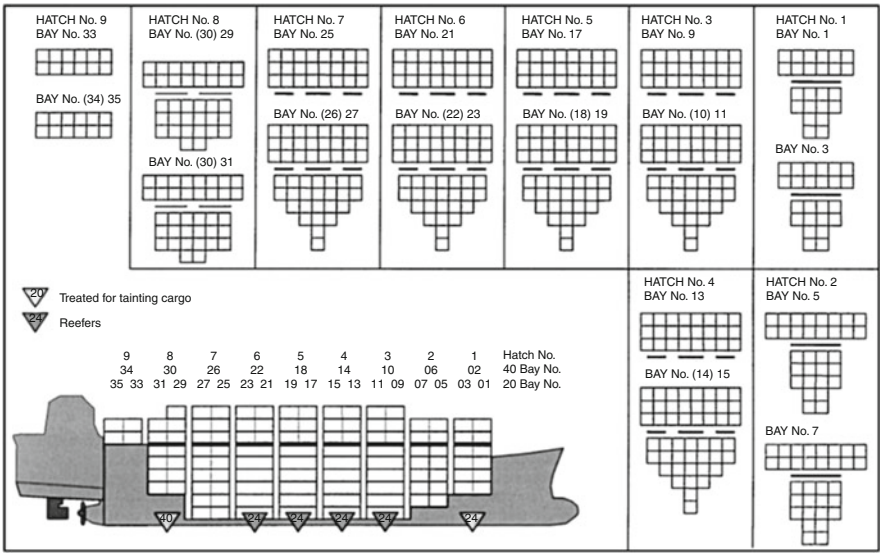


Fig. 1 General arrangement of cargo positioning (Wilson and Roach [18])

some empty containers that are loaded in the vessel in each port in the free spaces. This means that in each port the containers (demands) are unloaded and some empty containers are loaded in the same (now empty) slots in order to be returned to the origin port. Two major decisions must be considered: which ports should be visited by each vessel and the visit sequence, and how to stow the containers on board when considering all the placement requirements and LIFO strategy. The goals are to determine the best visiting sequence of ports and to reduce shifting of containers, while minimizing the total distribution cost.

As explained in [16] the short sea shipping main costs are related to the containers handling in the ports. A vessel carrying cargo to several ports may require a large number of shift operations. So the position of the containers on the vessels is crucial to reduce the time spent on a port to unload them. Taking this into account the vessel's stowage plan influences port handling, because the containers are loaded in vertical stacks located in different sections (bays) and the only way to access them is through the top of the stack. There are different handling operations of the containers:

1. Containers are loaded to be dispatched;
2. Containers are unloaded because they reached their destination;
3. Containers have to be shifted because they block the access to others that have to be unloaded, known as overstow;
4. Containers have to be re-positioned to improve the overall stowage (vessel stability) or to make the next port handling easier, which are known as re-handles

When a shift is performed, the container is always loaded in a different slot and the stowage plan is normally rearranged. Shifting is a time consuming activity, hence the arrangement of containers on board is crucial to achieve operational efficiency and reduce the number of shifts. To reduce the number of shifts, the LIFO strategy is considered when the routes are computed, which is part of the decision making process.

Beyond the costs implicit on a movement, there are other costs that have to be considered:

1. Costs related to ports, for example, taxes and utilization costs;
2. Costs related to vessels, for example the cost of a travel operation that depends on the cost of the fuel/mile and the cost of vessel utilization which depends on the crew member per day.

So, in order to reduce the global short sea distribution cost, an optimal route is computed that minimizes the total distribution costs. The CSSRP has several constraints that have to be considered. The constraints are divided in three groups: related to routes/ports (C1); related to vessels/loading (C2); and those that are related to both (C3).

(C1) The Routing constraints: Deadline constraint: deadline limits can be modeled as linear vehicle routing constraint, where the arrival time of the vessel to a port must be less than or equal to the smallest container deadline of the port demand.

(C2) The Loading and Placement constraints:

1. Vessel capacity constraints: maximal weight and number of containers. Weight limits and the number of slots can be modeled as a linear knapsack constraints, where the number of containers loaded and sum of its weights must be smaller or equal to the vessel capacity;
2. Positioning constraints, related to container placement stability in vessels. Can be modeled as a three-dimensional container loading problem constraints, where containers could be loaded directly on the vessel floor (or on the hatch cover) or above another container(s).

(C3) The Routing and Loading constraints: LIFO strategy, which is directly related to the number of shifts operations and it is used to bind the routing and the loading problem.

The challenge tackled in this study, is to develop a MIP model for the CSSRP that can achieve optimal solutions in a reduced computational time and can be applied to real problems, being the main contribution of the present work.

3 Literature Review

The CSSRP has received little attention in literature. All the approaches we were able to find have only considered the ships' container stowage problem (CSP), sometimes considering predefined port visiting sequence. Nevertheless, as far as we know the first published work that deals with the Container Stowage Problem (CSP) and the well-known Vehicle Routing Problem (VRP) with deadlines in an integrated way is from [14]. Later on, [16] proposes a MIP model that solves the CSSRP but with some assumptions in order to simplify the model. These simplifications are related to the containers. The distinction between containers (TEUs or FEUs and refrigerated) are not considered. In this model a demand to a given port is addressed like one big container with a defined dimension and a weight, that corresponds to the summation of dimensions and weights of all container's demand. So, the placement matrix is simplified and reduced. In this present work and with the aim to improve and adapt the model of [16] to real problems, these simplifications are not taken in consideration.

Regarding the ship routing and scheduling, [10] present an optimization-based solution approach for a real vessel's planning problem which the authors characterized as the Inventory Pickup and Delivery Problem with Time Windows (IPDPTW). The mathematical programming model was solved using the Dantzig–Wolfe decomposition, decomposing the original formulation into a sub-problem for each port and each ship. The linear programming relaxation of the master problem is solved by column generation, where the columns represent the vessel's routes or port's visit sequences. In order to make the integer solution optimal, the iterative solution process is embedded in a Branch-and-Bound search. Later on, [1] presented an integrated MIP model to solve the vessel's scheduling and

cargo-routing problems simultaneously. In this work a greedy heuristic, a column generation based algorithm and a two phase algorithm were developed. This approach is able to generate good schedules for vessels. Later on [14] proposes a genetic algorithm approach to improve the flexibility of short sea shipping and to increase its competitiveness with other means of freight transport. A logistic model was developed to manage a fleet of vessels which transport cargo to and from several ports, bearing in mind the cargo loading and delivery deadlines. The results show that the overall efficiency of short sea shipping can be improved.

In the last three decades unlike CSSRP, several works on CSP, also known as the Master Bay Problem (MBP), were published. Considering the description of the CSSRP in Sect. 2, is easily understandable that the CSP is a NP-hard problem [7]. This problem concerns the task of determining the arrangement of containers in a vessel. It can be categorized as an assignment problem where a set of containers with different characteristics and with a given port destination, must be assigned to slots in a vessel aiming to minimize the transportation cost. Since the late 1980s there were several works published about the CSP. To the best of our knowledge [4] was the first author to solve the stack overstowage problem using a dynamic programming algorithm. The goal was the arrangement policy and this approach was widely adopted in later works. Avriel and Penn [5] proposed the whole columns heuristic procedure to find the optimal solution for a stowage problem in a single rectangular bay with only accessibility constraints. This heuristic requires solving a ILP model after some pre-processing of the data. This method was proven to be limited because of the large number of binary variables and constraints needed to the formulation. Later on, [6] developed the suspensory heuristic procedure that achieves very satisfactory results in a short computation time. Nonetheless, the method proved to be very inflexible as far as the implementation of constraints is concerned. Binary linear programming formulations for the CSP with stability constraints, weight constraints, accessibility constraints, etc., can be found in [2, 9, 16] and [3]. In all of these works, the authors concluded that it is impossible to obtain optimal solutions through ILP for these problems with additional constraints. A very recent work that solves the vessel stowing planning, from [11] contradicts this statement. They decompose the problem and present a constraint programing and integer programming model for stowing a set of containers in a single bay section and solve real life problems to optimality in a reasonable computational time.

Several search methods such as Genetic Algorithms (GA), [12, 14], and tabu search [19] have also been applied to CSP. The advantage of using heuristic and meta-heuristic approaches to deal with this kind of problems has been proved with these works. Wilson and Roach [18] and later on [19], presented a two-phase method. In the first stage containers are grouped by their destination using a Branch-and-Bound search algorithm, aiming to reduce overstows and hatch movements at the next port-of-destination. After that, in the second stage, a tabu search algorithm is applied to the generalized solution, trying to move the containers and assigning them to a specific slot, in order to reduce re-handles, bearing in mind the stability constraints. The first time that the CSP was compared and characterized like another

well-known problem – the Bin Packing Problem – was by [17]. In this work the CSP is regarded as a bidimensional packing problem, where the bays of the vessels are regarded as bins, the number of slots in each bay is regarded as the capacity of the bins, and containers with different characteristics are treated as items to be packed. A two stage approach was developed: in the first stage two objective functions were considered, one to minimize the number of bays packed by containers and the other to minimize the number of overstows. Then the containers assigned to each bay in the first stage are allocated to special slots in the second stage, applying a tabu search algorithm. Constraints like weight, stability and overstows are considered.

4 The CSSRP Model Formulation

In this section a mixed integer programming model, aimed to guarantee the generation of optimal solutions to the CSSRP is presented. A mathematical model was developed to manage a fleet of vessels which must transport the container's demands to several ports, bearing in mind the cargo loading and delivery deadlines. The data and variables of the model are described in the following subsections.

4.1 General Model Components

The CSSRP is defined on a direct graph $G(P, A)$ where each port is represented by a node and has a different geographical location. $P = \{1, \ldots, p\}$ represents the set of ports and $A = \{(i, j) : i, j \in P, i \neq j\}$ the set of arcs in G. The length of each arc d_{ij} corresponds to the distance between port i and j in miles. For the distribution process, a vessel's fleet is available and represented by a set $V = \{1, \ldots, v\}$. For the sake of simplicity, it is assumed that the navigation speed (vel_k) is constant throughout all the arcs.

There are costs associated with ports and vessel's (Sect. 2). u_{ik} is the visiting cost (in euros) of port i by vessel k. This cost includes the tax and ports utilization costs, like for example, tugs and cranes. Let us assume that the initial port for vessel departure is port one and the visiting cost in this port is also considered. The costs related to the vessels are:

- c_k is the traveling operation cost of vessel k (per miles, in euros) which depends on the cost of fuel per mile;
- uc_k a vessel utilization cost (per day, in euros), which depends on the crew member number.

Each vessel k is characterized by its weight capacity $Wmax_k$ and the vessels capacity in number of containers $Cmax_k$.

In order to fulfill the capacity constraints of the vessels, another set of data related to the containers, must be presented. First we consider that all the containers have

standard dimensions (TEU and FEU). Second, we consider that there are two types of containers: the normal containers and the refrigerated containers. The initial port has a set of containers that must be delivered to other ports-of-destination. Each port-of-destination has a demand $td_i = dt_i + df_i + drt_i + drf_i$, composed of one or more types of containers, where dt_i, df_i, drt_i, drf_i are the number of normal containers of $20'$ and $40'$, refrigerated containers of $20'$ and $40'$, respectively. Also, for each port the demands' total weight given by $w_i = wt_i + wf_i + wrt_i + wrf_i$, where, wt_i, wf_i, wrt_i, wrf_i is the weight of all normal containers of $20'$ and $40'$, refrigerated containers of $20'$ and $40'$, for port i, respectively.

Moreover, the demand of each port is characterized by a delivery deadline dl_i. For simplicity's sake, all containers with the same destination have the same deadline that will be equal to the smallest containers' deadline for that port. In order to fulfill this constraint an estimation of the service time ts_{ik} of vessel k in port i must be considered and tt_{ijk} it's the required time that vessel k needs to transverse the arc (i, j), in days, given by (1):

$$\frac{d_{ij}}{vel_k \times 24} \tag{1}$$

The last set of data is related to the vessel loading and slots. As mentioned in Sect. 2 and due to the irregular configuration of the cargo space in a vessel, the bays $B = \{1, \ldots, b\}$ are defined by a matrix $posM_{kbcs}$ of slots (Fig. 2). The matrix has a set of rows (tier or cell) $C = \{1, \ldots, c\}$ and a set of columns (stacks) $S = \{1, \ldots, s\}$ Each cell of the matrix has assigned a value (0 or 1) that indicates if it is a vessels' slot or not. These slots can load one container of $40'$ or up to two containers of $20'$. Another matrix $refM_{kbcs}$ is used only to indicate the slots with electrical power (slots with value equal to 1), as mentioned before in Sect. 2, these slots are the only place where the refrigerated containers can be loaded (Fig. 3).

These slots can be occupied by one $40'$ or two $20'$ refrigerated containers, because there are two plugs available in each slot. It should be noted that, the normal containers can also be loaded in slots with plugs, as long as there aren't sufficient refrigerated containers to occupy them.

There is a cost related to the container's movement (load or unload) represented by *mc*, this cost is incurred for the handling operations of one single container.

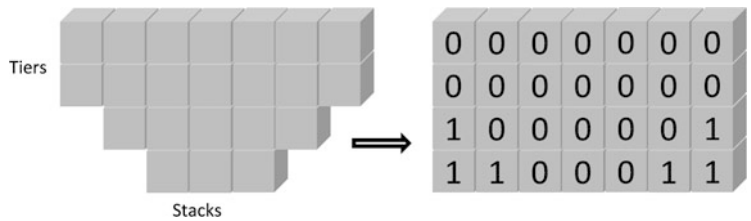


Fig. 2 Placement matrix

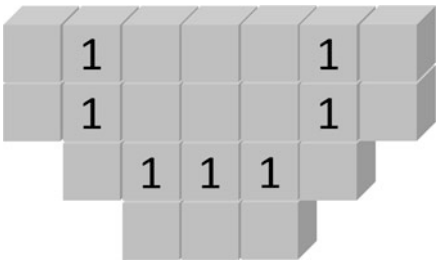


Fig. 3 Plugs location matrix

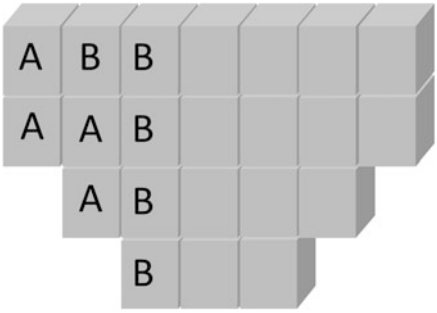


Fig. 4 Loading rule

4.2 Loading Rule

As in the three dimensional container loading problem (3D-CLP), there are several different ways to place the containers in vessels. In the 3D-CLP the boxes could be placed in the containers on the walls and layer building schemes, the boxes of the same type are arranged in rows or columns to fill one side or the floor of the container free space [8]. There are other ways to fill the containers, by homogeneous blocks, where each block consists of a set of equal box types, single columns or rows, etc.

In this problem, a loading rule was defined in order to reduce the overstows and related shift movements. This rule fills the bays row by row (stacks) from left to right and from bottom to top of each column (cells). Figure 4, shows the load rule application for a vessel where A and B represents the set of containers to delivery in port A and B.

This way the containers with a farthest destination are placed on the lowest part of the stack. This rule is achieved due to the order in which the indexes in the bays' matrixes positioning ($posM_{kbcs}$) are defined (see, Sect. 4.1) and the way in which the model is developed (see C3 – Loading and placement constraints, from 15 to 25 in Sect. 4.3). However there are some exceptions. Those are related to the refrigerated containers. In this case the containers are loaded in the existing plugged slots disregarding the loading rule.

4.3 The MIP Model

Several decisions and auxiliary variables were considered, some of them exclusively related to the routing problem, others to the container stowage problem:

1. x_{ijk} indicates if the vessel k traverses arc (i, j), or not.
2. s_{ij} gives us the arrival time of vessel k to port i.
3. $t_{kibcs}, f_{kibcs}, rt_{kibcs}, rf_{kibcs}$, these variables indicate if a type of container with destination port i is placed in slot (b, c, s) of vessel k or not. Those variables are related to the $20'$ containers t_{kibcs}, rt_{kibcs} vary between 0 and 2, because in one slot it could be loaded up to two containers of $20'$. On the other hand, the variables related to $40'$ containers f_{kibcs}, rf_{kibcs} vary between 0 and 1, because up to one container of $40'$ could be loaded in a slot.

Considering all the general model components, the objective function of the model is defined as:

$$Min \sum_{k \in V} \sum_{i \in P} \sum_{j \in P, j \neq i} x_{ijk} \left(d_{ij} c_k + u_{jk} + td_j mc \right) + \sum_{k \in V} uc_k s_{1k} \tag{2}$$

The CSSRP model minimizes the total route cost (2). This objective function is related to several constraints divided in three groups (as presented in Sect. 2):

C1 – Routing constraints:

$$\sum_{i \in P, i \neq j} x_{ijk} \leq 1, \quad \forall j \neq 1 \in P, \forall k \in V \tag{3}$$

$$\sum_{j \in P, j \neq i} x_{ijk} \leq 1, \quad \forall i \neq 1 \in P, \forall k \in V \tag{4}$$

$$\sum_{i \in P, i \neq j} x_{ijk} - x_{jik} = 0, \quad \forall k \in V, \forall j \in P \tag{5}$$

$$\sum_{j \in P} x_{1jk} \leq 1, \quad \forall k \in V \tag{6}$$

$$\sum_{k \in V} x_{ijk} \leq 1, \quad \forall i, j \in P \tag{7}$$

$$s_{ik} + ts_{ik} + tt_{ijk} \leq s_{jk} + M \left(1 - x_{ijk}\right), \quad \forall k \in V, \forall i, j \neq i \in P \tag{8}$$

$$s_{ik} \leq dl_i, \quad \forall k \in V, \quad \forall i \in P \tag{9}$$

This first set of constraints is related to the routing problem. Constraints (3), (4) and (5) are the flow conservation constraints. Equations (3) and (4) ensures that every port is visited only by one vessel. Equation (5) ensures that if a vessel arrives to a port it also leaves the same port. Equation (6) ensures that if a vessel is used, it

must begin its route in port one and Eq. (7) ensures that each port is visited only by one vessel. Constraint C1 (Sect. 2), is achieved with Eqs. (8) and (9). These ensure the feasibility of time scheduling defining the vessel setup time and the deadline constraint for serving a port, respectively. Constraint (8) guarantees that the port's service does not begin before the vessel's arrival to the port. This constraint uses a large multiplier (Big-M value) and in order to create valid inequalities we set $M = S_1^{max}$, where S_1^{max} is the maximum time needed to visit all ports. Constraint (9), guarantees that each container deadline is not violated.

C2 – The Loading and Placement constraints:

The following sets of constraints are related to positioning and loading constraints (C2, Sect. 2).

$$\sum_{i \in P} \sum_{j \in P, j \neq i} x_{ijk} \times w_j \leq Wmax_k, \quad \forall k \in V \tag{10}$$

$$\sum_{i \in P} \sum_{j \in P, j \neq i} x_{ijk} \times td_j \leq Cmax_k, \quad \forall k \in V \tag{11}$$

$$\sum_{k \in V} \sum_{b \in B} \sum_{c \in C} \sum_{s \in S} t_{kibcs} \leq dt_i, \quad \forall i \in P \tag{12}$$

$$\sum_{k \in V} \sum_{b \in B} \sum_{c \in C} \sum_{s \in S} f_{kibcs} \leq df_i, \quad \forall i \in P \tag{13}$$

$$\sum_{k \in V} \sum_{b \in B} \sum_{c \in C} \sum_{s \in S} rt_{kibcs} \leq drt_i, \quad \forall i \in P \tag{14}$$

$$\sum_{k \in V} \sum_{b \in B} \sum_{c \in C} \sum_{s \in S} rf_{kibcs} \leq drf_i, \quad \forall i \in P \tag{15}$$

Equations (10) and (11) are related to the vessels capacity. The first one states that the total demand on a route cannot exceed the vessel capacity in terms of weight and the second one in terms of number of containers. Equations (12), (13), (14) and (15), ensures that all demands are satisfied. Another set of constrains are related to containers' position on vessels. As mentioned before, there are two different matrixes, one related to the possible loading positing in a vessel and the other that indicates the slots with plugs.

$$t_{kibcs} + 2 \times posM_{kbcs} \leq 2, \quad \forall k \in V, \forall i \in P, \forall b \in B, \forall c \in C, \forall s \in S \tag{16}$$

$$f_{kibcs} + posM_{kbcs} \leq 1, \quad \forall k \in V, \forall i \in P, \forall b \in B, \forall c \in C, \forall s \in S \tag{17}$$

$$rt_{kibcs} + refM_{kbcs} \leq 2, \quad \forall k \in V, \forall i \in P, \forall b \in B, \forall c \in C, \forall s \in S \tag{18}$$

$$rf_{kibcs} + refM_{kbcs} \leq 1, \quad \forall k \in V, \forall i \in P, \forall b \in B, \forall c \in C, \forall s \in S \tag{19}$$

$$\sum_{i \in P} (t_{kibcs} + 2 \times f_{kibcs}) \leq 2, \quad \forall k \in V, \forall b \in B, \forall c \in C, \forall s \in S \tag{20}$$

$$\sum_{i \in P} (rt_{kibcs} + 2 \times rf_{kibcs}) \leq 2, \quad \forall k \in V, \forall b \in B, \forall c \in C, \forall s \in S \tag{21}$$

$$\sum_{i \in P} (t_{kibcs} + 2 \times rf_{kibcs}) \leq 2, \quad \forall k \in V, \forall b \in B, \forall c \in C, \forall s \in S \tag{22}$$

$$\sum_{i \in P} (t_{kibcs} + rt_{kibcs}) \leq 2, \quad \forall k \in V, \forall b \in B, \forall c \in C, \forall s \in S \tag{23}$$

$$\sum_{i \in P} (rt_{kibcs} + 2 \times f_{kibcs}) \leq 2, \quad \forall k \in V, \forall b \in B, \forall c \in C, \forall s \in S \tag{24}$$

$$\sum_{i \in P} (f_{kibcs} + rf_{kibcs}) \leq 1, \quad \forall k \in V, \forall b \in B, \forall c \in C, \forall s \in S \tag{25}$$

$$\sum_{i \in P} (rt_{kibcs} + t_{kibcs} + 2f_{kibcs} + 2posM_{kbcs}) \geq \tag{26}$$

$$\sum_{i \in P} (rt_{kibas} + t_{kibas} + 2f_{kibas} + 2posM_{kbas}), \tag{27}$$

$$\forall k \in V, \forall b \in B, \forall c, a < c \in C, \forall s \in S \tag{28}$$

For this reason a set of constrains, from (16) to (28), are needed in order to guarantee that the refrigerated containers are assigned only to slots with power supply ((18) and (19)) and also that in each one of these slots only one 40′ or up to two 20′ refrigerated containers can be placed (21). On the other hand, the normal containers can be placed in any slot in the vessel and each slot can have one 40′ container or up to two 20′ containers ((16) and (17)). The other set of constraints (20), (22), (23), (24) and (25) ensures that in any slot, it only could be loaded one 40′ container or one or two 20′ container, or it could be empty. Those constraints together with constraint (28), ensures the containers placement stability in vessels. These sets of constraints, give us the possible relative position between adjacent slots in the same stack. It guaranties that all the containers are fully supported. Nevertheless, it can never be placed any type of container in a slot if the slot below is empty or not fully occupied (Fig. 5).

C3 – Routing and Loading constraints:

$$\sum_{i \in P, i \neq j} x_{ijk} \times d_j = \sum_{b \in B} \sum_{c \in C} \sum_{s \in S} \left(t_{kjbcs} + f_{kjbcs} + rt_{kjbcs} + rf_{kjbcs}\right) \quad \forall k \in V, \forall j \in P \tag{29}$$

The two problems (route planning and container stowage) are integrated through Eq. (29), that binds the container loading variables to the vehicle routing problem variables, i.e., if a vessel visits a given port then the port's demand must be placed inside that vessel. To guarantee the LIFO strategy and to minimize the number of shifts (Routing and Loading Constraint – C3, Sect. 2), the loading rule (Sect. 4.2) is applied.

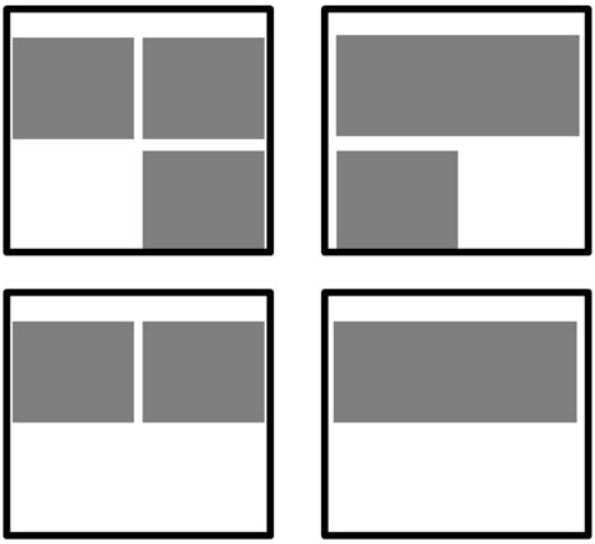


Fig. 5 Infeasible loading example

5 Computational Results

The main goal of this approach is to solve the short sea shipping distribution problem using an exact method that can achieve an optimal solution in a reasonable computational time. According to [11], a reasonable computation time for real problem solving applications are 15 minutes. The main idea is to apply this model to real-life problems taking in consideration that in the short sea distribution the average number of ports per route is five [14]. With this perspective, available data from real ships and ports was used. The problem instances and the model results are described in the following sections.

5.1 Problem Instances

The problem instances were developed based on the available data of a short sea distribution problem presented in [16], which considers two vessels and five ports and a homogeneous demand distribution per port. This means that the number of containers and weight for each port is similar and in some cases equal. These problem instances that we from now on call basic problems, were also developed taking into account [14] work. The scenarios utilized in [14] were collected in Porto Maritimo de Viana do Castelo, Portugal. In our problem instances, two different vessels were used. One of them also used in [16] and [14], the AXE that has a capacity of 348 TEUs, has 8 bays defined as a 9*8 matrix (cells*stacks). The other

vessel with a greater capacity that could achieve 5000 TEUs with a maximum bay size under deck of 69 TEUs. This vessel has 32 bays defined as a 14*15 matrix. Usually in short sea distribution, these kinds of vessels are not used, only the smaller types of vessels with around 348 TEUs. But we wanted to test the model with bigger vessels, in order to study its behavior when the number of slots is considerably increased.

It was considered that the fuel consumption and the velocity of the vessels are constant independently of the quantity and weight of the cargo. To prove the robustness of the model larger instances were generated, increasing the number of ports and using two vessels. In general, each port demand could not exceed the maximum capacity of the vessel in order to avoid the split of a port's demand between vessels. Also as in [16] it is assumed that all port tariffs are the same and some costs, such as insurance costs and others, were neglected. These cost simplifications only imply changes in the objective function value. Then, in order to study the model behavior another set of instances were developed. The idea is to study the effect of the demand quantities and the containers deadline variation, in terms of CPU time and optimal solutions. For this reason, in all instances, the quantity of containers per type in each demand varies upon: at least 50 % for the 20′ normal containers; at least 35 % for the 40′ normal containers; between 0 % and 10 % for the 20′ refrigerator containers and between 0 % and 5 % for the 40′ refrigerator containers. The problem instances were solved by CPLEX software, and the experiments were run on an Intel CORE i7 vPro 2.2 GHz with 8 Gb of memory.

5.2 Model Results and Sensibility Analyses

For each type of vessel, keeping the same demand distribution per port and the same type of deadlines, like in the basic problems, 10 problem instances were tested. The difference between those instances are the number of ports and the number of vessels. The results achieved are presented in Table 1.

For the instances with one vessel, except for 15 ports, the model always achieved an optimal solution. All the CPU times are very small, except for the 15 ports and 2 vessels problem instance (almost 2 hours). For the problems with one vessel, increasing the number of ports results in a small increase in the computation time. However there is a more significant rise in the computational time when, for the same number of ports the number of vessels is increased. This was expected due to the number of variables related to the containers' positioning. In order to give an idea of the model size relevance to the difficulty of solving some of the instances, for the smallest and biggest problems the numbers of constraints and variables are presented in Table 2:

As can be seen in Table 2, there is a large increase in the number of constraints and variables in the instances of Panamax vessel with 15 ports and 2 vessels. This

Table 1 Problem instances results

Vessels	N. vessels	N. ports	Obj. function	CPU (sec)	GAP (%)
AXE	1	5	321,084	0.55	0
AXE	2	5	413,676	1.81	21.74
AXE	1	10	534,155	0.61	0
AXE	2	10	685,386	16.97	3.2
AXE	1	15	630,386	66.89	0.80
AXE	2	15	754,303	6400.08	27.58
Panamax	1	5	797,743	0.69	0
Panamax	2	5	1,034,385	37.13	14.85
Panamax	1	10	1,324,606	0.70	0
Panamax	2	10	1,513,677	754.42	3.94

Table 2 Problem size

		N. constraints	N. variables
AXE: 8 bays	5 ports, 2 vessels	24,770	17,345
AXE: 8 bays	15 ports, 2 vessels	60,060	52,325
Panamax: 36 bays	5 ports, 2 vessels	806,578	268,922
Panamax: 36 bays	15 ports, 2 vessels	2,420,468	807,352

implies an incredible increase of the computational time. This is the reason why the results of Panamax vessel instances with 15 ports are not presented. Taking into account the majority of the tested instances we can conclude that the integrality GAP decreases with the increase of the number of ports. The opposite behavior happens when varying the number of vessels, in particular for 15 ports. Nevertheless, and according to the problems sizes (Table 2), with the results achieved it is proven that this model can be applied to real problems of short sea distribution (characterized by a reduced number of ports and small vessels), due to the small computational time needed to achieve an optimal solution.

As mentioned before, a sensibility analyses was performed with different instances. Our intent was to explore the effect of varying the quantities of the demands and deadlines. So, in order to prove the robustness and behavior of the model, three different scenarios only for 5 ports problem instances, were tested.

SA2: Varying the demand distribution per port: The behavior of the model and the effect of varying the demand distribution per port was studied (Table 3). As in [16], three types of demands' distribution were considered: weak heterogeneous and strong heterogeneous. As explained before, in the so called basic problems, the load distribution is based on real data and the container's number and weight is very similar sometimes equal, between each port. On the other hand, a weak heterogeneous distribution happens when there are more significant differences between them. Furthermore, strong heterogeneous is when differences between the demands of each port are very significant, for example: a demand from one port could be composed by only 10 containers and for another by 1200.

Table 3 Computational results varying the demands and deadlines

Vessels	N. ports/vessels	Problem type	Obj. function	CPU (sec)	GAP (%)
AXE	5/1	Basic Problem	321,084	0.55	0
AXE	5/1	Strong Het. Demands	321,079	0.51	0
AXE	5/1	Weakly Het. Demands	321,073	0.52	0
AXE	5/1	Narrow Deadlines	472,522	0.51	0
AXE	5/1	Wide Deadlines	272,077	0.58	0
AXE	5/2	Basic Problem	413,676	1.81	21.74
AXE	5/2	Strong Het. Demands	413,741	1.34	22.86
AXE	5/2	Weakly Het. Demands	413,671	1.03	0
AXE	5/2	Narrow Deadlines	413,676	1.72	0
AXE	5/2	Wide Deadlines	413,676	2.68	14.55
Panamax	5/1	Basic Problem	797,743	0.69	0
Panamax	5/1	Strong Het. Demands	797,743	0.60	0
Panamax	5/1	Weakly Het. Demands	797,689	1.34	0
Panamax	5/1	Narrow Deadlines	1,171,162	0.67	0
Panamax	5/1	Wide Deadlines	676,899	1.14	0
Panamax	5/2	Basic Problem	1,034,385	37.13	14.85
Panamax	5/2	Strong Het. Demands	965,051	32.54	30.39
Panamax	5/2	Weakly Het. Demands	1,033,867	28.63	34.83
Panamax	5/2	Narrow Deadlines	1,034,385	53.52	0
Panamax	5/2	Wide Deadlines	1,034,385	85.50	17.63

SA3: Varying the deadlines: Another test was made in order to see the behavior of the model according to different types of deadlines. In this case, narrow and wide deadlines were tested using the basic problems load distribution (Table 3). A deadline is called narrow, when it is almost the same amount of time required to traverse the arc. On the other hand, when a deadline is called wide it is because those values are large enough that it makes it so (for some or for the majority of the ports) there are no deadline constraints.

These two data types could be critical because, for example, if the demand to a given port is equal to or very close to the vessel capacity, that vessel will visit only that port, making the problem potentially easier to solve due to the decrease in the number of possible route/loading combinations. The effects on the CPU time and optimal solutions, on varying the demands distribution and deadlines are presented in Table 3. The results obtained when the optimal solution is achieved, problem instances with one vessel, denotes a relationship between the type of demand and the computational time needed to solve the problem. For strong heterogeneous distribution the computational time is always smaller than considering the other two load distributions. Moreover, for problems with two vessels, the highest computational time is always associated with basic problems distribution. Related to the deadlines variation and for the problem instances with one vessel, the optimal solution was always achieved. In these cases and for wide

deadlines the computational time is always longer than the computational time of the basic problems. But decreasing the size of the deadlines (narrow deadlines) it could be seen that the computational time decreases and in a significant way compared to the wide deadlines. We think that this fact can be explained with the same arguments presented above for the demands variation. Namely, if a port has a deadline that is equal to the travel time that the vessels needs to get there, so this port must be the next port to visit independently if it is the farthest port or not. On the other hand, with wide deadlines and in the problem instance with one AXE vessel, it could be seen that the optimal solution was significantly reduced. This could be explained due to the fact of the size of deadlines be such as, that this constraint no longer exists and the route is the shortest one. Another curiosity is related to the problem instance with two Panamax vessels. In this case, unlike previously, the model only achieved the optimal solution with the narrow deadlines.

6 Conclusions

In this work a Mixed Integer Programing model to solve the short sea distribution problem named as container stowage and ship routing problem (CSSRP), was presented. This problem can be approached like the integration of two well-known NP-Hard problems: the Vehicle Routing Problem and the 3-Dimensional Loading Problem. However in CSSRP the demands have deadlines and the containers placement locations are fixed. Besides, in CSSRP it must be decided where to load the demands to a given port in order to reduce the unloading time and the number of shifts movements. The CSSRP is also a NP-hard problem. Despite the complexity of the problem and of the presented model, the results obtained for the set of problems tested prove that this model can be applied to solve real life short sea shipping problems. It was proven that the model in the most instances reaches an optimal integer solution, without unnecessary movements of cargo (shifts) during the trips, within a very short computational time. In addition, the results showed that the computational time to get the optimal solution, besides depending on number of ports to visit and number of vessels, also depends on the type of demands distribution and deadlines. The developed approach in this work shows that the solutions quality of the short sea shipping problem can be solved and improved by having an integrated perspective of the two above mentioned problems (the VRP problem and the CSP problem).

Acknowledgements This work was supported by Portuguese funds through the CIDMA – Center for Research and Development in Mathematics and Applications, and the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia), within project PEst-OE/MAT/UI4106/2014.

References

1. Agarwal, R., Ergun, O.: Ship scheduling and network design for cargo routing in linear shipping. Transp. Sci. **42**(2), 175–196 (2008)
2. Ambrosino, D., Sciomachen, A., Tanfani, E.: Stowing a containership: the master bay plan problem. Transp. Res. Part A **38**, 81–99 (2004)
3. Ambrosino, D., Sciomachen, A., Tanfani, E.: A decomposition heuristics for the container ship stowage problem. J. Heuristics **12**, 211–233 (2006)
4. Aslidis, A.H.: Combinatorial algorithms for stacking problems. PhD dissertation, MIT (2000)
5. Avriel, M., Penn, M.: Exact and approximate solutions of the container ship stowage problem. Comput. Ind. Eng. **25**, 271–274 (1993)
6. Avriel, M., Penn, M., Shpirer, N., Witteboon, S.: Stowage planning for container ships to reduce the number of shifts. Ann. Oper. Res. **76**, 55–71 (1998)
7. Avriel, M., Penn, M., Shpirer, N.: Container ship stowage problem: complexity and connection to the coloring of circle graphs. Discret. Appl. Math. **103**, 271–279 (2000)
8. Bortfeldt, A., Wascher, G.: Constraints in container loading a state-of-the-art review. Eur. J. Oper. Res. **229**, 1–20 (2013)
9. Botter, R.C., Brinati, M.A.: Stowage container planning: a model for getting an optimal solution. In: Computer Applications in the Automation of Shipyard Operation and Ship Design, VII, pp. 217–229. North-Holland, Amsterdam/New York (1992)
10. Christiansen, M., Nygreen, B.: A method for solving ship routing problems with inventory constraints. Ann. Oper. Res. **81**, 357–378 (1998)
11. Delgado, A., Jensen, R.M., Janstrup, K., Rose, T.H., Andersen, K.H.: A Constraint Programming model for fast optimal stowage of container vessel bays. Eur. J. Oper. Res. **220**, 251–261 (2012)
12. Dubrovsky, O., Levitin, O.G., Penn, M.: A genetic algorithm with a compact solution encoding for the container ship stowage problem. J. Heuristics **8**, 585–599 (2002)
13. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search algorithm for a routing and container loading problem. Transp. Sci. **9**(3), 342–350 (2006)
14. Martins, T., Moura, A., Campos, A.A., Lobo, V.: Genetic algorithms approach for containerships fleet management dependent on cargo and their deadlines. In: Proceedings of IAME 2010: Annual Conference of the International Association of Maritime Economists, Lisbon 7–9 July 2010
15. Moura, A., Oliveira, J.F.: An integrated approach to the vehicle routing and container loading problems. Oper. Res. Spectr. **31**, 775–800 (2009)
16. Moura, A., Oliveira, J., Pimentel, C.: A mathematical model for the container stowage and ship routing problem. J. Math. Model. Algorithms Oper. Res. **12**(3), 217–231 (2013)
17. Wei-Ying, Z., Yan, L., Zhuo-Shang, J.T.: Model and algorithm for container ship stowage planning based on Bin-packing problem. J. Marine Sci. Appl. **4**(3), 30–36 (2005)
18. Wilson, I.D., Roach, P.A.: Principles of combinatorial optimization applied to container-ship stowage planning. J. Heuristics **5**(4), 403–418(16) (1999)
19. Wilson, I.D., Roach, P.A.: Container stowage planning: a methodology for generating computerized solutions. J. Oper. Res. Soc. **51**, 1248–1255 (2000)

A Consumption-Investment Problem with a Diminishing Basket of Goods

Abdelrahim S. Mousa, Diogo Pinheiro, and Alberto A. Pinto

Abstract We consider the problem faced by an economic agent trying to find the optimal strategies for the joint management of her consumption from a basket of K goods that may become unavailable for consumption from some random time τ_i onwards, and her investment portfolio in a financial market model comprised of one risk-free security and an arbitrary number of risky securities driven by a multi-dimensional Brownian motion. We apply previous abstract results on stochastic optimal control problem with multiple random time horizons to obtain a sequence of dynamic programming principles and the corresponding Hamilton-Jacobi-Bellman equations. We then proceed with a numerical study of the value function and corresponding optimal strategies for the problem under consideration in the case of discounted constant relative risk aversion utility functions (CRRA).

1 Introduction

The origin of Optimal Control is related with Calculus of Variations, which started to be developed to deal with problems arising in Physics from the middle of 1600s onwards. One of the main techniques used to address optimal control problems is the dynamic programming principle, introduced by Bellman [1–3] in 1950s. The dynamic programming principle has been extended to address stochastic optimal control problems [4, 14, 15, 18], providing a backwards recursive relation for the value function associated with such problems and, under additional conditions, a nonlinear partial differential equation known as the Hamilton-Jacobi-Bellman

A.S. Mousa (✉)
Faculty of Science, Department of Mathematics, Birzeit University, Ramallah, Palestine
e-mail: asaid@birzeit.edu

D. Pinheiro
Department of Mathematics, Brooklyn College of the City University of New York, NY, USA
e-mail: dpinheiro@brooklyn.cuny.edu

A.A. Pinto
Faculty of Science, LIAAD – INESC TEC and Department of Mathematics, University of Porto, Rua do Campo Alegre, 687, 4169-007, Portugal
e-mail: aapinto@fc.up.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_17

(HJB) equation. For further details on this subject we refer to the textbooks by Fleming and Rishel [12], Fleming and Soner [13] and Yong and Zhou [21]. Another key technique to address optimal control problems is the Pontryagin's maximum principle [9, 10]. For further details on this approach see Haussmann [16], Bismut [5–7] and Yong and Zhou [21].

More recently there has been considerable interest in the study of optimization problems with an objective functional and state variable dynamics depending on a random time horizon. Such problems were often considered in the context of mathematical finance. See, for instance, the papers [8, 11, 20] for further details.

A general framework for this class of problems is considered in [19]. In that paper a family of stochastic optimal control problems is considered with the property that the objective functional depends on multiple random time horizons. The state variable follows a stochastic differential equation driven by a standard multi-dimensional Brownian motion. For that class of stochastic optimal control problems, a sequence of dynamic programming principles and the corresponding Hamilton-Jacobi-Bellman equations are derived. In the current paper we discuss an application of the abstract results of [19] to a consumption-investment problem with a diminishing basket of goods.

We consider an economic agent investing in a financial market model consisting of one risk-free security and an arbitrary number of risky securities driven by a multi-dimensional Brownian motion. We assume that the economic agent is consuming from a basket of K goods that may become unavailable for consumption from some random time τ_i onwards. The economic agent goal is to jointly maximize: (i) the utility derived from consumption of the goods available at each instant of time; and (ii) the utility derived from wealth at some multiple random instants of time, representing either the moment at which a given good becomes unavailable or the deterministic horizon T.

This paper is organized as follows. In Sect. 2 we review the main results in [19], i.e. the sequence of dynamic programming principles and corresponding Hamilton-Jacobi-Bellman equations (HJB). In Sect. 3 we set up the optimization problem under consideration here by introducing the underlying financial market and the corresponding wealth process. We restate the stochastic optimal control problem under consideration as one with a fixed planning horizon, and derive a sequence of dynamic programming principles and the corresponding Hamilton-Jacobi-Bellman equations. We use the results discussed in Sect. 2 to proceed with detailed analysis of our consumption problem with multiple random time horizons. We conclude in Sect. 4.

2 Preliminary Results

In this section we review the problem formulation of [19]. Let $K \in \mathbb{N}$ be fixed and $T > 0$ be a finite time horizon. Assume that $(\Omega, \mathscr{F}, \mathbb{P})$ is a complete probability space equipped with a filtration $\mathbb{F} = \{\mathscr{F}_t, t \in [0, T]\}$ given by the $\mathbb{P}$-augmentation

of the filtration generated by the M-dimensional Brownian motion $W(t)$, $\sigma\{W(s), s \leq t\}$ for $t \geq 0$.

We denote by $L^1_{\mathscr{F}_t}([0,T];\mathbb{R})$ the set of all $\{\mathscr{F}_t\}_{t\geq 0}$-adapted $\mathbb{R}$-valued processes $x(\cdot)$ such that

$$E\left[\int_0^T |x(t)|\,\mathrm{d}t\right] < \infty$$

and by $L^1_{\mathscr{F}_T}(\Omega;\mathbb{R})$ the set of $\mathbb{R}$-valued $\mathscr{F}_T$-measurable random variables X such that $E[|X|]$ is finite. The spaces $L^1_{\mathscr{F}_t}([0,T];\mathbb{R})$ and $L^1_{\mathscr{F}_T}(\Omega;\mathbb{R})$ are defined on the filtered probability space $(\Omega, \mathscr{F}, \mathbb{F}, \mathbb{P})$.

Assume that $\tau_1, \tau_2, \ldots, \tau_K$ are non-negative independent and identically distributed continuous random variables defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$, with distribution function $F : [0,\infty) \to [0,1]$ and density $f : [0,\infty) \to \mathbb{R}^+$. Moreover, assume that the random variables τ_i are completely independent of the filtration $\mathbb{F}$ for all $i \in \{1,2,\ldots,K\}$.

Let $\tau_{(1)} \leq \tau_{(2)} \leq \ldots \leq \tau_{(K)}$ be the order statistics associated with the random variables $\tau_1, \tau_2, \ldots, \tau_K$. Define the following finite sequence of times:

(i) the sequence $(\xi_i)_{i=1}^K$ is given by

$$\xi_i = \min\{\tau_i, T\}, \qquad i = 1,2,\ldots,K\,.$$

(ii) the sequence $(t_i)_{i=0}^K$ is given by $t_0 = 0$ and

$$t_i = \min\{\tau_{(i)}, T\}, \qquad i = 1,2,\ldots,K\,.$$

Using the sequence (t_i), we introduce the sequence of random intervals Δ_k as being given by

$$\Delta_k = [t_{k-1}, t_k)\,, \qquad k \in \{1,2,\ldots,K\}\,.$$

For each $t \in [0, t_K)$, let $I(t)$ be the set of indices given by

$$I(t) = \{i \in \{1,2,\ldots,K\} : \tau_i > t\}$$

and let $\kappa : [0,t_K) \to \{1,2,\ldots,K\}$ be the function assigning to each $t \in [0,t_K)$ the cardinality of the set $I(t)$.

We now introduce some technical assumptions:

(A1) (U,d) is a Polish space.

(A2) The maps $f_k : [0,T] \times \mathbb{R}^N \times U \to \mathbb{R}^N$, $g_k : [0,T] \times \mathbb{R}^N \times U \to \mathbb{R}^{N\times M}$, $\Psi_k : [0,T] \times \mathbb{R}^N \to \mathbb{R}$ and $L_k : [0,T] \times \mathbb{R}^N \times U \to \mathbb{R}$, $k = 1,2,\ldots,K$, are uniformly continuous with respect to all its variables, Lipschitz continuous with respect to the variable x, and bounded when restricted to $x = 0$.

Let $f : [0, t_K] \times \mathbb{R}^N \times U \to \mathbb{R}^N$ and $g : [0, t_K] \times \mathbb{R}^N \times U \to \mathbb{R}^{N\times M}$ be the functions defined through the relations

$$f(t, x, u) = f_k(t, x, u) \qquad \text{and} \qquad g(t, x, u) = g_k(t, x, u)$$

for every $(t, x, u) \in \Delta_k \times \mathbb{R}^N \times U$, extended by continuity to $t = t_K$. Consider the stochastic controlled system

$$\begin{aligned} \mathrm{d}x(t) &= f\left(t, x(t), u(t)\right) \mathrm{d}t + g\left(t, x(t), u(t)\right) \mathrm{d}W(t)\,, \qquad t \in [0, t_K]\,, \\ x(0) &= x_0 \end{aligned}$$

and the objective functional

$$J(u(\cdot)) = E\left[\sum_{i=1}^{K}\left(\int_0^{\xi_i} L_i\left(t, x(t), u(t)\right) \mathrm{d}t + \Psi_i\left(\xi_i, x(\xi_i)\right)\right)\right],$$

where $x(t)$ is a state trajectory in $\mathbb{R}^N$ with the corresponding control $u(t) \in U$, $L_i(\cdot, x(\cdot), u(\cdot)) \in L^1_{\mathscr{F}_t}([0, t_K]; \mathbb{R})$ and $\Psi_i(\xi_i, x(\xi_i)) \in L^1_{\mathscr{F}_{\xi_i}}(\Omega; \mathbb{R})$, $i = 1, 2, \ldots, K$.

To use the dynamic programming methods, for any initial condition $(s, y) \in [0, t_K) \times \mathbb{R}^N$, we consider the state equation:

$$\begin{aligned} \mathrm{d}x(t) &= f\left(t, x(t), u(t)\right) \mathrm{d}t + g\left(t, x(t), u(t)\right) \mathrm{d}W(t)\,, \qquad t \in [s, t_K]\,, \\ x(s) &= y\,, \end{aligned} \tag{1}$$

along with the objective functional

$$J(s, y; u(\cdot)) = E\left[\sum_{i\in I(s)} \int_s^{\xi_i} L_i\left(t, x(t), u(t)\right) \mathrm{d}t + \Psi_i\left(\xi_i, x(\xi_i)\right)\right].$$

For each $s \in [0, t_K)$, we denote by $\mathscr{U}^w[s, T]$ the set of all 5-tuples $(\Omega, \mathscr{F}, \mathbb{P}, W(\cdot), u(\cdot))$ for which the following conditions hold:

(i) $(\Omega, \mathscr{F}, \mathbb{P})$ is a complete probability space;
(ii) $\{W(t)\}_{t\geq s}$ is an M-dimensional standard Brownian motion defined on $(\Omega, \mathscr{F}, \mathbb{P})$ over $[s, T]$, and $\mathscr{F}_t^s$ is the filtration generated by $\{W(t)\}_{t\geq s}$, $\sigma\{W(r) : s \leq r \leq t\}$, augmented by all the $\mathbb{P}$-null sets in $\mathscr{F}$;
(iii) $u : [s, t_K] \times \Omega \to U$ is an $\{\mathscr{F}_t^s\}_{t\geq s}$-adapted process on $(\Omega, \mathscr{F}, \mathbb{P})$;
(iv) under $u(\cdot)$, for any $y \in \mathbb{R}^N$ Eq. (1) admits a unique solution $x(\cdot)$ on $(\Omega, \mathscr{F}, \{\mathscr{F}_t^s\}_{t\geq s}, \mathbb{P})$.

We call $\mathscr{U}^w[s, T]$ the set of weak admissible controls.

Using the notations introduced above, the goal is to find a weak admissible control $u(\cdot) \in \mathscr{U}^w[s,T]$ such that the following identity holds

$$V(s,y) = \sup_{u(\cdot)\in\mathscr{U}^w[s,T]} J(s,y;u(\cdot)), \qquad (s,y) \in [0,t_K) \times \mathbb{R}^N .$$

The function $V(s,y)$ defined above is referred to as the value function.

For every $0 \le s \le t$ and every $i,k \in \{1,2,\ldots,K\}$ such that $i \ge k$, let $G^{\pm}_{\tau_{(i)}|\tau_{(k)}}(t,s)$ denote the conditional probabilities

$$G^{+}_{\tau_{(i)}|\tau_{(k)}}(t,s) = P\Big(\{\tau_{(i)} > t\} \mid \{\tau_{(k)} > s\}\Big)$$
$$G^{-}_{\tau_{(i)}|\tau_{(k)}}(t,s) = P\Big(\{\tau_{(i)} \le t\} \mid \{\tau_{(k)} > s\}\Big)$$

and let $g^{-}_{\tau_{(i)}|\tau_{(k)}}(t,s)$ denote the density function of $G^{-}_{\tau_{(i)}|\tau_{(k)}}(t,s)$, given by

$$g^{-}_{\tau_{(i)}|\tau_{(k)}}(t,s) = \frac{\mathrm{d}}{\mathrm{d}t}\, G^{-}_{\tau_{(i)}|\tau_{(k)}}(t,s) .$$

Furthermore, for every $k \in \{1,2,\ldots,K\}$ such that $t_{k-1} < T$, $i \in \{1,2,\ldots,K\}$ such that $i \ge k$, $j \in I(t_{k-1})$ and $t_{k-1} \le s \le t$, define the *conditional Lagrangian function* to be

$$\mathscr{L}^{j}_{\tau_{(i)}|\tau_{(k)}}(t,s,x,u) = G^{+}_{\tau_{(i)}|\tau_{(k)}}(t,s) L_j(t,x,u) + g^{-}_{\tau_{(i)}|\tau_{(k)}}(t,s)\Psi_j(t,x) .$$

We now state the two main results of [19].

Theorem 1 (Dynamic programming principle) *Assume that conditions (A1) and (A2) hold. Then, for every $k \in \{1,2,\ldots,K\}$ such that $t_{k-1} < T$, the restriction of the value function V to the set $\Delta_k \times \mathbb{R}^N$ is identically equal to the function determined by the recursive relation*

$$V(s,y) = \sup_{u(\cdot)\in\mathscr{U}^w[s,T]} E\Bigg[G^{+}_{\tau_{(k)}|\tau_{(k)}}(\hat{s},s) V(\hat{s}, x(\hat{s};s,y,u(\cdot))) + \frac{1}{\kappa(t_{k-1})} \sum_{i=k}^{K} \sum_{j\in I(t_{k-1})} \int_s^{\hat{s}} \mathscr{L}^{j}_{\tau_{(i)}|\tau_{(k)}}(t,s,x(t;s,y,u(\cdot)),u(t))\mathrm{d}t \,\Big|\, s \in \Delta_k \Bigg], \quad \hat{s} \in [s,T]$$

combined with the boundary condition

$$V(T,x) = \sum_{j\in I(t_{k-1})} \Psi_j(T,x) .$$

Let $I \subseteq \mathbb{R}$ be an interval and denote by $C^{1,2}\left(I \times \mathbb{R}^N; \mathbb{R}\right)$ the set of all continuous functions $V : I \times \mathbb{R}^N \to \mathbb{R}$ such that V_t, V_x, and V_{xx} are all continuous functions of (t, x). Using the sequence of Dynamic programming principles from the previous theorem, we were able to derive the associated sequence of Hamilton-Jacobi-Belman equations.

Theorem 2 *Suppose that conditions (A1) and (A2) hold. Additionally, assume that for every $i, j \in \{1, 2, \ldots, K\}$ such that $i \geq j$ the conditional density functions $g^{-}_{\tau(i)|\tau(j)}(t, s)$ are uniformly continuous with respect to t.*

Let $k \in \{1, 2, \ldots, K\}$ be such that $t_{k-1} < T$ and assume that the value function V is such that $V \in C^{1,2}\left(\Delta_k \times \mathbb{R}^N; \mathbb{R}\right)$. Then, on each set $\Delta_k \times \mathbb{R}^N$ the value function V is identically equal to the solution of the Hamilton-Jacobi-Bellman equation

$$\begin{cases} V_t - g^{-}_{\tau(k)|\tau(k)}(t,t)V + \sup\limits_{u \in U} \mathscr{H}^k(t, x, u, V_x, V_{xx}) = 0 \\ V(T, x) = \sum_{j \in I(t_{k-1})} \Psi_j(T, x) \,, \qquad\qquad (t, x) \in [t_{k-1}, T) \times \mathbb{R}^N \,, \end{cases}$$

where the Hamiltonian function $\mathscr{H}^k$ is given by

$$\begin{aligned} \mathscr{H}^k(t, x, u, p, B) = & \sum_{j \in I(t_{k-1})} \Big(L_j(t, x, u) + G_k(t)\Psi_j(t, x) \Big) \\ & + \langle p, f_k(t, x, u) \rangle + \frac{1}{2} tr\left(g_k^T(t, x, u) B g_k(t, x, u) \right) \end{aligned}$$

for all $(t, x, u, p, B) \in [0, T] \times \mathbb{R}^N \times U \times \mathbb{R}^N \times \mathscr{S}^N$, $\mathscr{S}^N$ is the set of all $N \times N$ symmetric real matrices and $G_k(t)$ is given by

$$G_k(t) = \frac{1}{\kappa(t_{k-1})} \sum_{i=k}^{K} g^{-}_{\tau(i)|\tau(k)}(t, t) \,.$$

3 An Application to a Consumption-Investment Decision Problem

Consider a financial market consisting of one risk-free security and a fixed number $N \geq 1$ of risky securities. The asset prices $(S_0(t))_{0 \leq t \leq T}$ and $(S_n(t))_{0 \leq t \leq T}$, $n = 1, \ldots, N$, evolve according to the differential equations:

$$\begin{aligned} \mathrm{d}S_0(t) &= r(t)S_0(t)\mathrm{d}t \,, & S_0(0) &= s_0 > 0 \,, \\ \mathrm{d}S_n(t) &= \mu_n(t)S_n(t)\mathrm{d}t + S_n(t) \sum_{m=1}^{M} \sigma_{nm}(t)\mathrm{d}W_m(t) \,, & S_n(0) &= s_n > 0 \,, \end{aligned}$$

where $W(t) = (W_1(t), \ldots, W_M(t))^T$ is a standard M-dimensional Brownian motion on a filtered complete probability space $(\Omega, \mathscr{F}, \mathbb{F}, \mathbb{P})$, $r(t)$ is the riskless interest rate, $\mu(t) = (\mu_1(t), \ldots, \mu_N(t)) \in \mathbb{R}^N$ is the vector of the risky-assets appreciation rates and $\sigma(t) = (\sigma_{nm}(t))_{1 \leq n \leq N, 1 \leq m \leq M}$ is the matrix of risky-assets volatilities. Each sub-σ-algebra $\mathscr{F}_t$ represents the information available to any financial market observer during the time span $[0, t]$.

For simplicity of exposition, we assume that the coefficients $r(t)$, $\mu(t)$ and $\sigma(t)$ are deterministic continuous functions on the interval $[0, T]$. We also assume that the interest rate $r(t)$ is positive for all $t \in [0, T]$ and the matrix $\sigma(t)$ is such that $\sigma\sigma^T$ is non-singular for Lebesgue almost all $t \in [0, T]$ and satisfies the following integrability condition

$$\sum_{n=1}^{N} \sum_{m=1}^{M} \int_0^T \sigma_{nm}^2(t) \mathrm{d}t < \infty \, .$$

Furthermore, we suppose that there exists an $(\mathscr{F}_t)_{0 \leq t \leq T}$-progressively measurable process $\pi(t) \in \mathbb{R}^M$, called the market price of risk, such that for Lebesgue-almost-every $t \in [0, T]$ the risk premium

$$\alpha(t) = (\mu_1(t) - r(t), \ldots, \mu_N(t) - r(t)) \in \mathbb{R}^N$$

is related to $\pi(t)$ by the equation

$$\alpha^T(t) = \sigma(t)\pi^T(t) \qquad \text{a.s.}$$

and the following two conditions hold

$$\int_0^T \|\pi(t)\|_M^2 < \infty \qquad \text{a.s.}$$

$$E\left[\exp\left(-\int_0^T \pi(s)\mathrm{d}W(s) - \frac{1}{2}\int_0^T \|\pi(s)\|_M^2 \, \mathrm{d}s\right)\right] = 1 \, ,$$

where $\|\cdot\|_M$ denotes the Euclidean norm in $\mathbb{R}^M$. The existence of $\pi(t)$ ensures the absence of arbitrage opportunities in the financial market defined above. See [17] for further details on market viability.

Let $\tau_1, \tau_2, \ldots, \tau_K$ be non-negative independent and identically distributed continuous random variables defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$. Moreover, assume that $\tau_1, \tau_2, \ldots, \tau_K$ are independent of the filtration $\mathbb{F}$ generated by the Brownian motion $W(t)$. Let (ξ_i) and (t_i) be as defined in Sect. 2.

We will consider the problem faced by an economic agent investing in the financial market described above and consuming from a basket of K goods that may

become unavailable for consumption from some time τ_i onwards. The economic agent goal is to jointly maximize:

(i) the utility derived from consumption of the goods available at each instant of time;
(ii) the utility derived from wealth at the multiple random instants of time $\xi_1, \xi_2, \ldots, \xi_K$, representing either the moment at which a given good becomes unavailable or the deterministic horizon T.

For each $i = 1, 2, \ldots, K$ and $t \in [0, t_K)$, let $c_i(t)$ denote the economic agent consumption of the ith good at the instant of time t. The *consumption process* is given by

$$c(t) = (c_1(t), \ldots, c_K(t)) .$$

We assume that the consumption process is a $(\mathscr{F}_t)$-progressively measurable non-negative process with the property that each component $c_i(\cdot)$ is non-negative and satisfies the following integrability condition

$$\int_0^{\xi_i} \|c(t)\|_K^2 \, \mathrm{d}t < \infty \qquad \text{a.s.} .$$

For each $n = 0, 1, \ldots, N$ and $t \in [0, t_K]$, let $\theta_n(t)$ denote the fraction of the economic agent wealth allocated to the asset S_n at time t. The *portfolio process* is given by

$$\Theta(t) = (\theta_0(t), \theta_1(t), \cdots, \theta_N(t)) \in \mathbb{R}^{N+1} ,$$

where

$$\sum_{n=0}^{N} \theta_n(t) = 1 , \qquad 0 \le t \le t_K . \tag{2}$$

We assume that the portfolio process is $(\mathscr{F}_t)$-progressively measurable and that

$$\int_0^{t_K} \|\Theta(t)\|_{N+1}^2 \, \mathrm{d}t < \infty \qquad \text{a.s.} .$$

Using relation (2), we can always write $\theta_0(t)$ in terms of $\theta_1(t), \ldots, \theta_N(t)$. From now on, we will define the portfolio process in terms of the *reduced portfolio process* $\theta(t) \in \mathbb{R}^N$ given by

$$\theta(t) = (\theta_1(t), \theta_2(t), \cdots, \theta_N(t)) \in \mathbb{R}^N .$$

We define the *wealth process* $X(t)$, for $t \in [0, t_K]$ by

$$X(t) = x - \sum_{i=1}^{K} \int_0^{\xi_i \wedge t} c_i(s)\, \mathrm{d}s + \sum_{n=0}^{N} \int_0^t \frac{\theta_n(s) X(s)}{S_n(s)}\, \mathrm{d}S_n(s)\ ,$$

where x is the economic agent initial wealth. The last equation can be rewritten in the differential form

$$\begin{aligned} \mathrm{d}X(t) = &\left(- \sum_{i \in I(t)} c_i(t) + \left(\theta_0(t) r(t) + \sum_{n=1}^{N} \theta_n(t) \mu_n(t) \right) X(t) \right) \mathrm{d}t \\ &+ \sum_{n=1}^{N} \theta_n(t) X(t) \sum_{m=1}^{M} \sigma_{nm}(t) \mathrm{d}W_m(t)\ , \end{aligned}$$

where $I(t)$ is the index set defined in Sect. 2.

Let us denote by $\mathscr{A}(x)$ the set of all admissible decision strategies, i.e. all admissible choices for the control variables $(c, \theta) \in \mathbb{R}^{K+N}$. The dependence of $\mathscr{A}(x)$ on x denotes the restriction imposed on the wealth process by the boundary condition $X(0) = x$. Similarly, let us denote by $\mathscr{A}(t, x)$ the set of all admissible decision strategies (c, θ) for the dynamics of the wealth process with boundary condition $X(t) = x$.

The economic agent problem is to find a consumption-investment strategy $(c, \theta) \in \mathscr{A}(x)$ which maximizes the expected utility

$$V(x) = \sup_{(c,\theta) \in \mathscr{A}(x)} E\left[\sum_{i=1}^{K} \int_0^{\xi_i} L_i\left(t, c_i(t)\right) \mathrm{d}t + \Psi_i\left(\xi_i, X(\xi_i)\right) \right],$$

where $L_i\left(t, c_i(t)\right)$, $i = 1, 2, \ldots, K$, is the utility functions describing the economic agent preferences regarding the ith good consumption over the random time interval $[0, \xi_i]$, and $\Psi_i(\xi_i, X(\xi_i))$, $i = 1, 2, \ldots, K$, are the utility functions describing the economic agent preferences concerning the amount of wealth held at the instants of time $\xi_1, \xi_2, \ldots, \xi_K$.

Assume that the conditions of Theorem 2 hold. Then, for every $k \in \{1, 2, \ldots, K\}$ such that $t_{k-1} < T$ the value function V restricted to the set $\Delta_k \times \mathbb{R}^N$ is identically equal to the solution of the Hamilton-Jacobi-Bellman equation:

$$\begin{cases} V_t - g^{-}_{\tau_{(k)}|\tau_{(k)}}(t,t) V + \sup\limits_{(c,\theta) \in \mathscr{A}(t,x)} \mathscr{H}^k(t, x, c, \theta, V_x, V_{xx}) = 0 \\ V(T, x) = \sum_{i \in I(t_{k-1})} \Psi_i(T, x)\ , \qquad (t, x) \in [t_{k-1}, T) \times \mathbb{R}^N\ , \end{cases} \tag{3}$$

where the Hamiltonian function $\mathscr{H}^k$ is given by

$$\begin{aligned}\mathscr{H}^k(t,x,c,\theta,p,B) = &\sum_{i\in I(t_{k-1})}\left(L_i(t,c_i)+G_k\left(t\right)\Psi_i(t,x)\right)\\ &+\left(-\sum_{i\in I(t_{k-1})}c_i(t)+\left(r(t)+\sum_{n=1}^{N}\theta_n(\mu_n(t)-r(t))\right)x\right)p\\ &+\frac{x^2}{2}\sum_{m=1}^{M}\left(\sum_{n=1}^{N}\theta_n\sigma_{nm}(t)\right)^2 B\end{aligned}$$

and $G_k\left(t\right)$ is as given in the statement of Theorem 2.

3.1 Constant Relative Risk Aversion Utility Functions

We will now focus our attention on the class of discounted CRRA utility functions, given by

$$L_i\left(t,c_i\right)=\mathrm{e}^{-\rho t}\frac{c_i^{\gamma_i}}{\gamma_i}\,,\qquad \Psi_i(t,x)=\mathrm{e}^{-\rho t}\frac{x^{\beta_i}}{\beta_i}\,,\qquad i=1,2,\ldots,K, \tag{4}$$

where the risk aversion parameters γ_i and β_i are different from zero and strictly less than one for every $i\in\{1,2,\ldots,K\}$, and the discount rate ρ is positive.

Computing the first-order conditions for a regular interior maximum of $\mathscr{H}^k$ with respect to (c,θ), we obtain the following two equalities:

$$\begin{aligned}-V_x(t,x)+\frac{\mathrm{d}L_i}{\mathrm{d}c_i}(t,c_i^*)=0\,,&\qquad i\in I(t_{k-1})\\ xV_x\left(t,x\right)\alpha+x^2V_{xx}\left(t,x\right)\theta^*\sigma\sigma^T=0_{\mathbb{R}^N}\,.&\end{aligned}$$

Thus, solving with respect to c and θ, we are able to express the optimal strategies in terms of the value function derivatives as

$$\begin{aligned}c_i^*(t,x)&=\left(e^{\rho t}V_x(t,x)\right)^{-1/(1-\gamma_i)}\,,\qquad i\in I(t_{k-1})\\ \theta^*(t)&=-\frac{V_x(t,x)}{xV_{xx}(t,x)}\alpha(t)\xi\,,\end{aligned} \tag{5}$$

where ξ denotes the non-singular square matrix $(\sigma\sigma^T)^{-1}$. Using the utility functions (4) and substituting c_i^* and θ^* in the HJB equation (3), we arrive at the following

partial differential equation

$$\begin{aligned} & V_t(t,x) - g^-_{\tau(k)|\tau(k)}(t,t)V(t,x) + r\,(t)\,xV_x(t,x) \\ & - \Sigma(t)\frac{(V_x(t,x))^2}{V_{xx}(t,x)} + G_k\,(t)\,\mathrm{e}^{-\rho t}\sum_{i\in I(t_{k-1})}\frac{x^{\beta_i}}{\beta_i} \\ & + \sum_{i\in I(t_{k-1})}\left(\frac{1-\gamma_i}{\gamma_i}\right)e^{-\rho t/(1-\gamma_i)}(V_x(t,x))^{-\gamma_i/(1-\gamma_i)} = 0\,, \end{aligned} \tag{6}$$

where $\Sigma(t)$ is given by

$$\Sigma(t) = \alpha(t)\xi\alpha^T(t) - \frac{1}{2}\left\|\sigma^T(t)\xi\alpha^T(t)\right\|^2_M \tag{7}$$

and the terminal condition is given by

$$V(T,x) = \sum_{i\in I(t_{k-1})}\Psi_i(T,x)\,. \tag{8}$$

The following result provides the optimal strategies for discounted CRRA utility functions in case where the risk aversion parameters are all equal. Quite naturally, we obtain that the optimal strategy consists in consuming identical amounts of each good available for consumption at a given time.

Corollary 1 *Let ξ denote the non-singular square matrix given by $(\sigma\sigma^T)^{-1}$. Assume that the risk aversion parameters γ_i and β_i are all equal. Then, for every $k \in \{1,2,\ldots,K\}$ such that $t_{k-1} < T$ the value function V restricted to the set $\Delta_k \times \mathbb{R}$ is of the form*

$$V(t,x) = \kappa(t_{k-1})a_k(t)\frac{x^\gamma}{\gamma}\,,$$

where $a_k(t)$ is the solution of the boundary value problem

$$\begin{aligned} & \frac{da_k(t)}{dt} + \Phi_k\,(t)\,a_k(t) + \eta(t)\,(\kappa(t_{k-1})a_k\,(t))^{-\gamma/(1-\gamma)} + G_k\,(t)\,\mathrm{e}^{-\rho t} = 0 \\ & a_k(T) = e^{-\rho T}\,, \end{aligned}$$

the function $\Sigma(t)$ is as given in (7), $G_k\,(t)$ is as given in the statement of Theorem 2 and $\Phi_k(t)$ and $\eta(t)$ are given by

$$\begin{aligned} \Phi_k(t) &= \gamma\left(r(t) + \frac{\Sigma(t)}{1-\gamma}\right) - g^-_{\tau(k)|\tau(k)}\,(t,t)\,, \\ \eta(t) &= \gamma\,(1-\gamma)\,e^{-\rho t/(1-\gamma)}\,. \end{aligned}$$

Furthermore, on each set $\Delta_k \times \mathbb{R}$, the optimal strategies are given by

$$c_i^*(t,x) = x\left(\kappa(t_{k-1})e^{\rho t}a_k(t)\right)^{-1/(1-\gamma)}, \qquad i \in I(t_{k-1})$$
$$\theta^*(t,x) = \frac{1}{1-\gamma}\,\alpha(t)\xi\,.$$

Proof Start by noting that in the case where the risk aversion parameters γ_i and β_i are all equal (we denote such common value by γ), the partial differential equation (6) becomes

$$V_t(t,x) - g^-_{\tau_{(k)}|\tau_{(k)}}(t,t)V(t,x) + r(t)\,xV_x(t,x) - \Sigma(t)\frac{(V_x(t,x))^2}{V_{xx}(t,x)} \tag{9}$$
$$+ \sum_{i\in I(t_{k-1})} \frac{1}{\gamma}\left(G_k(t)\,\mathrm{e}^{-\rho t}x^{\lambda} + (1-\gamma)\,e^{-\rho t/(1-\gamma)}\,(V_x(t,x))^{-\gamma/(1-\gamma)}\right) = 0\,,$$

where $\Sigma(t)$ is as given in (7), $G_k(t)$ is as given in the statement of Theorem 2 and the terminal condition is as given in (8).

We consider an ansatz of the form

$$V(t,x) = \sum_{i\in I(t_{k-1})} a_k(t)\frac{x^{\gamma}}{\gamma} = \kappa(t_{k-1})a_k(t)\frac{x^{\gamma}}{\gamma}$$

and substitute it in (9). A simple computation implies that $a_k(t)$ is determined by the following boundary value problem

$$\frac{\mathrm{d}a_k(t)}{\mathrm{d}t} + \Phi_k(t)\,a_k(t) + \eta(t)\,(\kappa(t_{k-1})a_k(t))^{-\gamma/(1-\gamma)} + G_k(t)\,\mathrm{e}^{-\rho t} = 0$$
$$a_k(T) = e^{-\rho T}\,,$$

where $\Phi_k(t)$ and $\eta(t)$ are as given in the statement of this corollary. Furthermore, using (5) we obtain that optimal strategies on $\Delta_k \times \mathbb{R}$ are given by

$$c_i^*(t,x) = x\left(\kappa(t_{k-1})e^{\rho t}a_k(t)\right)^{-1/(1-\gamma)}, \qquad i \in I(t_{k-1})$$
$$\theta^*(t,x) = \frac{1}{1-\gamma}\,\alpha(t)\xi\,,$$

concluding the proof.

3.2 Numerical Solution

Before concluding, we provide a numerical example where the risk aversion parameters do not coincide. We take the deterministic fixed horizon to be $T = 20$ and consider a financial market with one risk-free security and one risky asset ($N = 1$) driven by a one-dimensional Brownian motion ($M = 1$). The riskless interest rate is $r = 0.04$, the risky asset appreciation rate is $\mu = 0.08$ and its volatility is $\sigma = 0.19$, whereas the discount rate is taken to be $\rho = 0.04$. We assume that there is a basket of three goods available for consumption ($K = 3$) up until some random times τ_1, τ_2, τ_3, which are assumed to be independent and identically distributed non-negative random variables with distribution function given by

$$F(t) = 1 - \exp\left(-0.0010748t - 0.0000035t^2\right) , \qquad t \geq 0 .$$

Finally, the economic agent preferences are described by CRRA utilities of the form (4) with risk aversion parameters $\gamma_1 = 0.5$, $\gamma_2 = 0.45$, $\gamma_3 = 0.4$ and $\beta_1 = \beta_2 = \beta_3 = 0.5$.

In Fig. 1, we plot the value function $V(t, x)$ for a realization where $\tau_1 = 10$, $\tau_2 = 15$ and $\tau_3 \geq 20$. The value function is obtained by solving three HJB equations of the form (6), restricting its solutions to the sets $\Delta_i \times \mathbb{R}^+$, $i \in \{1, 2, 3\}$, and finally, combining the corresponding graphs into a single one. Note the existence

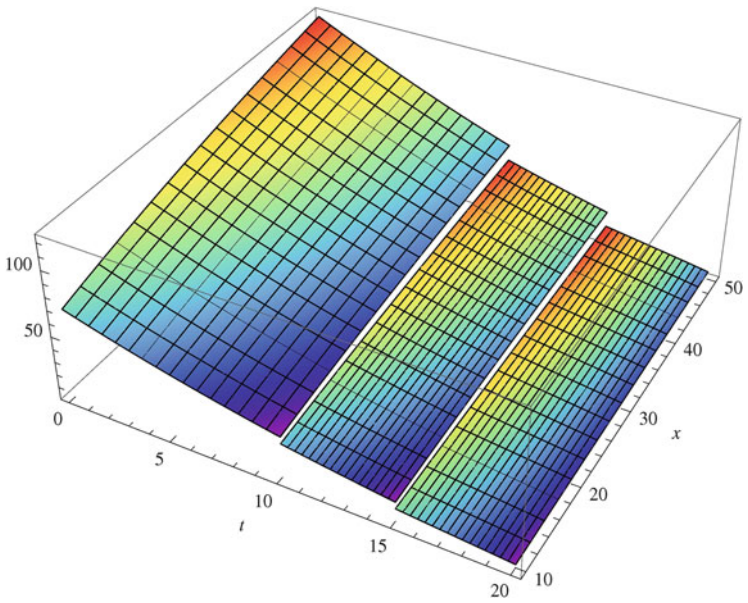


Fig. 1 Plot of the value function $V(t, x)$ for values $(t, x) \in [0, 20] \times [10, 50]$ and a realization where $\tau_1 = 10$, $\tau_2 = 15$ and $\tau_3 \geq 20$

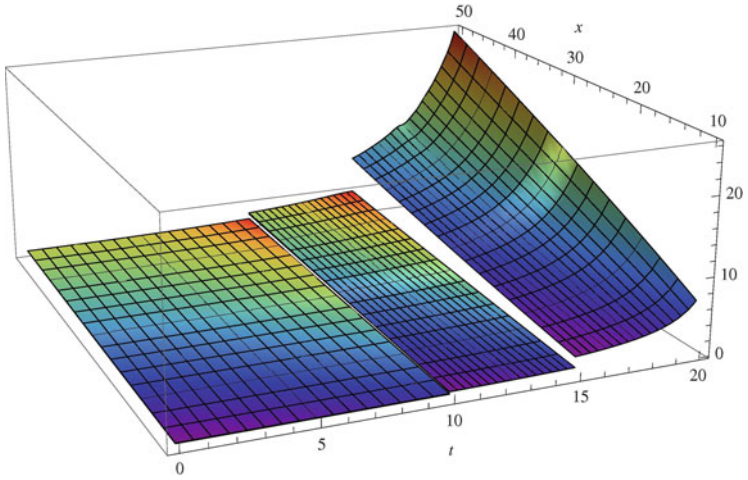


Fig. 2 Plot of the optimal consumptions $c^*(t,x)$ for values $(t,x) \in [0,20] \times [10,50]$ and a realization where $\tau_1 = 10$, $\tau_2 = 15$ and $\tau_3 \geq 20$

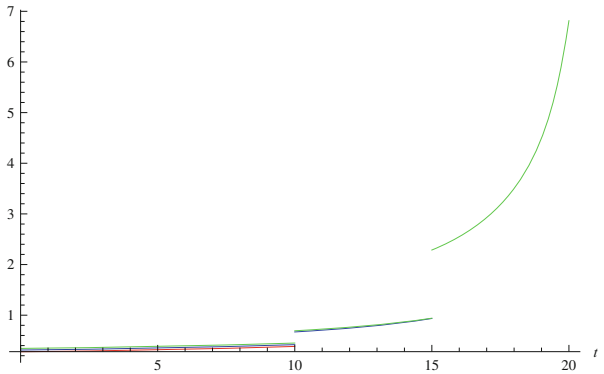


Fig. 3 Optimal consumptions c^* for $x = 10$

of discontinuities of the value function V on the sets $t_1 = 10$ and $t_2 = 15$. Such behaviour is due to the strong dependence of the functional J on the random variables τ_1, τ_2, τ_3.

In Fig. 2, we plot the optimal consumptions $c_i^*(t,x)$, $i \in \{1,2,3\}$, for a realization where $\tau_1 = 10$, $\tau_2 = 15$ and $\tau_3 \geq 20$. The optimal consumptions $c_i^*(t,x)$ are obtained directly from the value function according to (5). Note that on the set $\Delta_1 \times \mathbb{R}^+$ there exist three overlapping graphs, whereas on $\Delta_2 \times \mathbb{R}^+$ there are only two, and on $\Delta_3 \times \mathbb{R}^+$ only one. This is made clear in Figs. 3 and 4, where a section of Fig. 2 is presented for a fixed value $x = 10$. Note that the optimal consumptions are increasing functions of time t. Moreover, as should be expected, the optimal consumptions of the remaining goods jump up when one of the goods becomes

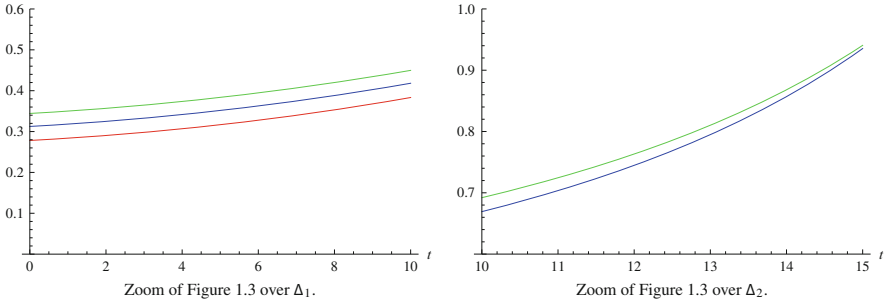


Fig. 4 A section of the optimal consumptions for $x = 10$. The optimal consumption c_1^* is plotted in *red*, while c_2^* is plotted in *blue* and c_3^* is plotted in *green*

unavailable, since the same amount of wealth can be distributed among a smaller number of goods.

4 Conclusions

We have studied a consumption-investment problem for an economic agent who is (i) consuming from a basket of K goods that may become unavailable for consumption from some random time τ_i onwards, and (ii) investing her saving in a financial market consisting of one risk-free security and a fixed number of risky securities with diffusive terms driven by a standard multi-dimensional Brownian motion. We have applied the main abstract results of [19] to describe the value function associated with the stochastic optimal control problem under consideration here. We have then discussed the properties of the optimal strategies in the case where the economic agent has the same discounted CRRA utility functions for consumption and terminal wealth.

Acknowledgements A. A. Pinto thanks the financial support of LIAAD–INESC TEC through program PEst, USP-UP project, Faculty of Sciences, University of Porto, Calouste Gulbenkian Foundation, FEDER and COMPETE Programmes, and Fundação para a Ciência e a Tecnologia (FCT) through the Project "Dynamics and Applications" (PTDC/MAT/121107/2010). A. S. Mousa thanks Birzeit University and its Department of Mathematics.

References

1. Bellman, R.E.: On the theory of dynamic programming. Proc. Natl. Acad. Sci. USA **38**, 716–719 (1952)
2. Bellman, R.E.: An introduction to the theory of dynamic programming. Rand Corporation Report pp. R–245 (1953)

3. Bellman, R.E.: Dynamic programming and a new formalism in the calculus of variations. Proc. Natl. Acad. Sci. USA **40**, 231–235 (1954)
4. Bellman, R.E.: Dynamic programming and stochastic control process. Inf. Control **1**, 228–239 (1958)
5. Bismut, J.M.: Analyse convexe et probabilités, Thèse. PhD thesis, Faculté des Sciences de Paris, Paris (1973)
6. Bismut, J.M.: Théorie probabiliste du contrôle des diffusions. Mem. Am. Math. Soc. **4**, 1–130 (1976)
7. Bismut, J.M.: An introductory approach to duality in optimal stochastic control. SIAM Rev. **20**, 62–78 (1978)
8. Blanchet-Scalliet, C., El Karoui, N., Jeanblanc, M., Martellini, L.: Optimal investment decisions when time-horizon is uncertain. J. Math. Econ. **44**(11), 1100–1113 (2008)
9. Boltyanski, V.G., Gamkrelidze, R.V., Pontryagin, L.S.: On the theory of optimal processes. Dokl. Akad. Nauk SSSR **10**, 7–10 (1956)
10. Boltyanski, V.G., Gamkrelidze, R.V., Pontryagin, L.S.: The theory of optimal processes, I: the maximum principle. Am. Math. Soc. Transl. **2**, 341–382 (1961)
11. Duarte, I., Pinheiro, D., Pinto, A.A., Pliska, S.R.: Optimal life insurance purchase, consumption and investment on a financial market with multi-dimensional diffusive terms. Optimization **63**, 1737–1760 (2014)
12. Fleming, W.H., Rishel, R.W.: Deterministic and Stochastic Optimal Control. Springer, New York (1975)
13. Fleming, W.H., Soner, H.M.: Controlled Markov Processes and Viscosity Solutions, 2nd edn. Springer, New York (2006)
14. Florentin, J.J.: Optimal control of continuous time, markov, stochastic systems. J. Electron. Control **10**, 473–488 (1961)
15. Florentin, J.J.: Partial observability and optimal control. J. Electron. Control **13**, 263–279 (1962)
16. Haussmann, U.G.: General necessary conditions for optimal control of stochastic systems. Math. Prog. Study **6**, 34–48 (1976)
17. Karatzas, I., Shreve, S.: Methods of Mathematical Finance. Springer, New York (1998)
18. Kushner, H.J.: Optimal stochastic control. IRE Trans. Auto. Control **AC-7**, 120–122 (1962)
19. Mousa, A.S., Pinheiro, D., Pinto, A.A.: A family of stochastic optimal control problems with multiple random time horizons (2014, submitted)
20. Pliska, S.R., Ye, J.: Optimal life insurance purchase and consumption/investment under uncertain lifetime. J. Bank. Financ. **31**, 1307–1319 (2007)
21. Yong, J., Zhou, X.Y.: Stochastic Controls: Hamiltonian Systems and HJB Equations. Springer, New York (1999)

Assessing Technical and Economic Efficiency of the Artisanal Dredge Fleet in the Portuguese West Coast

M.M. Oliveira, A.S. Camanho, and M.B. Gaspar

Abstract The bivalve dredge fleet is by far the most extensively studied fleet among the Portuguese artisanal segment. It is considered one of the most important artisanal fisheries, essentially due to the number of fishermen and vessels involved and to the high volume and value of the catches. The present study aimed to explore the efficiency of the dredge fleets that operated in the west coast of Portugal between 2006 and 2012. The methodology was based on the use of data envelopment analysis to assess vessels' efficiency. The inputs considered included the number of days at sea, a biomass stock indicator, and the characteristics of the vessels (power, length and tonnage). The annual fishing quota per vessel was also included in the model as a contextual factor. In the technical efficiency analysis, the outputs were defined by the weight of captures for three different bivalve species. Using data on the prices of each species in the wholesale market, revenue efficiency was also estimated to complement the technical efficiency analysis. The results allowed to gain insights concerning the performance of both Northwest and Southwest fleets, considering both technical and economic aspects of the fishery. It was also possible to identify the benchmark vessels, whose practices should be followed by the other vessels of the fleet.

M.M. Oliveira (✉)
Instituto Português do Mar e da Atmosfera I.P./IPMA, Avenida 5 de Outubro, 8700-305 Olhão, Portugal

Centro de Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, Bloco C6, Piso 4, Campo Grande, 1749-016 Lisboa, Portugal
e-mail: moliveira@ipma.pt

A.S. Camanho
Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
e-mail: acamanho@fe.up.pt

M.B. Gaspar
Instituto Português do Mar e da Atmosfera I.P./IPMA, Avenida 5 de Outubro, 8700-305 Olhão, Portugal

Centro de Ciências do Mar, Universidade do Algarve, Campus de Gambelas, 8005-139 Faro, Portugal
e-mail: mbgaspar@ipma.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_18

1 Introduction

The estimation of a Decision Making Unit (DMU) efficiency, according to Farrel [9] can be based on a comparison between observed and optimal values of production (outputs), given the resources consumed (inputs). This author distinguished two components of efficiency: technical and allocative. In fisheries, the first component can be interpreted as the ability of a vessel to obtain maximal catch from a given set of inputs (e.g. vessel's characteristics, fishing days, crew, and fuel consumption), whereas the second component reflects the ability of a vessel to use the outputs in optimal proportions, given their respective prices and the production technology. These measures can be combined to provide a measure of economic efficiency (also called revenue efficiency when an output orientation is adopted for the assessment). From this perspective, revenue efficiency can be defined as the ability of a vessel to maximise the revenue obtained, given the inputs consumed, the value of the catches and the features of the production technology. Hence, efficiency analysis in fisheries is an asset that contributes to the sector sustainability by guiding managerial decision making. The efficiency studies require data on input and output factors that are frequently not available for artisanal fisheries (the lack of data is an unsolved issue, with important consequences in this context (Guyader et al. [10])). The factors most frequently used as inputs are vessel characteristics, fishing effort, operational costs and stock abundance indices (e.g. Sharma and Leung [27]; Kirkley et al. [14, 15]; Pascoe and Coglan [23]). Concerning the output factors, most studies in both single and multispecies fisheries use the landed weight of the catches (e.g. Sharma et al. [28]; Pascoe and Herrero [24]; Hoff [12]; Lindebo et al. [18]) or the value of the catches (e.g. Tingley et al. [32]; Maravelias and Tsitsika [19]; Idda et al. [13]). As argued by Herrero and Pascoe [11], in single-species fisheries, the weight and value of the catches are quite often proportional, resulting in similar efficiency measures whilst in multispecies the use of weight and/or value of catches leads to different results, and thus should be selected in accordance with the purpose of the analysis. An efficiency assessment can be performed with different methodologies. Data Envelopment Analysis (DEA), a nonparametric, linear programming method, is the most frequently used in fisheries due to its characteristics. This method constructs an envelopment production frontier which maps out the greatest output for a given level of input, such that all observed points lie on or below this frontier. The production frontier (also known as "best practice frontier") is formed by the efficient DMUs. The efficiency of the remaining DMUs is measured by the distance to this frontier. Measuring efficiency with DEA allows the analyst to incorporate multiple inputs and outputs directly in the analysis, and does not require the specification of a structural relationship between the inputs and the outputs, leading to greater flexibility in the frontier estimation. Therefore, the DEA approach has been successfully applied in fisheries in order to assess technical efficiency (e.g. Dupont et al. [7]; Kirkley et al. [16]; Tingley et al. [31]; Vestergaard et al. [33]; Pascoe and Herrero [24]; Tingley and Pascoe [29, 30]), revenue efficiency (e.g. Lindebo et al. [18]; Oliveira et al. [20, 22]), profit efficiency (e.g. Pascoe and

Tingley [25]) and cost efficiency (e.g. Alam and Murshed-e-Jahan [1]). The present study explores the technical, allocative and revenue efficiency of the artisanal bivalve dredge fleets that operated in the west coast of Portugal (Northwest and Southwest) between 2006 and 2012. The main purpose of this analysis was to identify the best practices to be followed in both fleets, including the specification of most appropriate features of the vessels in terms of inputs. The efficiency of each vessel was estimated with DEA, considering fixed inputs (vessel power, length, tonnage, and an indicator of stock biomass) and a variable input (number of days at sea). An annual quota per vessel was also included in the model as a contextual factor. Revenue efficiency was estimated as a complement to the technical efficiency, using price data for each species in the wholesale market. A two-dimensional graphical representation of vessel's performance enabled us to identify the benchmark vessels, both in terms of those that maximized the weight of the catch for the species landed, given their inputs, as well as the vessels that selected the most appropriate target species to maximize the revenue of the fishing activity, given output prices. The definition of targets for inefficient vessels was also addressed, corresponding to the values of the catch for each species that would maximize the revenue.

2 Methodology

The DEA models were first proposed by Charnes et al. [5]. In the last three decades, several models were developed, covering a broad range of issues. In the present study, it was used a formulation with upper bound constraints on the outputs (see Cooper et al. [6], p. 224), which is particularly useful for evaluations involving maximum levels of outputs, as is the case of quota managed fisheries. In a quota managed fishery, a vessel may not be able to expand output fully because the catches are capped by regulation. Thus, the quota should be considered as an upper bound for the output variables representing the weight of catches.

Consider n DMUs, denoted by DMU_j for $j = 1, \ldots, n$, which use the inputs x_{ij}, where $(x_{1j}, \ldots, x_{mj}) \in \mathbb{R}^m_+$, to obtain the outputs y_{rj} $(y_{1j}, \ldots, y_{sj}) \in \mathbb{R}^s_+$. Assume that the maximum value of the sum of all outputs is bounded by the quota limit Q_{j_0} for each DMU_{j_0}. The efficiency of each DMU_{j_0} is given by the reciprocal of the factor (δ) by which the outputs of the DMU_{j_0} can be expanded, obtained from the following model based on Cooper et al. [6]:

$$
\begin{aligned}
& max\{\delta \mid \\
& x_{ij_0} \geq \sum_{j=1}^{n} \lambda_j x_{ij}, \quad i = 1, \ldots, m \\
& \delta y_{rj_0} \leq \sum_{j=1}^{n} \lambda_j y_{rj}, \quad r = 1, \ldots, s
\end{aligned} \tag{1}
$$

$$\sum_{r=1}^{s}\sum_{j=1}^{n}\lambda_j y_{rj} \le Q_{j_0},$$

$$\lambda_j \ge 0, \forall j\}$$

Model (2) is an output oriented model and assumes the existence of constant returns to scale (CRS). The value of $1/\delta^*$ is the measure of technical efficiency (TE) of DMU_{j_0}. Comparing the formulation of model (2) and the one proposed by Cooper et al. [6], the main difference resides is that the bounds are not specified for each output considered individually, but are instead specified in terms of the total weight of captures allowed for each vessel. An important by-product of the efficiency assessments concerns the specification of peers (i.e. benchmarks) and targets for inefficient units. The benchmarks for the DMU_{j_0} under assessment are the units with values of λ_j^* greater than zero in the optimal solution to model (2). Since the vessels were analysed with an output oriented perspective, the estimation of output targets for each DMU_{j_0} is particularly important. The targets corresponding to both radial and non-radial expansion of the outputs leading to efficient operation in the Pareto-Koopmans sense are obtained as shown in (2). According to Koopmans ([17], p.60), a producer is technical efficient if an increase in any output requires a reduction in at least one other output or an increase in at least one input. For further details on the Pareto-Koopmans definition of efficiency (see Cooper et al. [6], p.45).

$$y_{rj_0}^{target} = \sum_{j=1}^{n}\lambda_j^* y_{rj}, \quad r = 1,\ldots,s. \tag{2}$$

DEA model was also used to estimate economic efficiency, following Farrell [9] concepts. This author described a revenue maximization assessment, corresponding to the assumption that the DMUs intend to maximise the revenue obtained, given the resources consumed and the value of the output prices. In order to obtain a measure of revenue efficiency, the maximum revenue that can be obtained by DMU_{j_0}, given the current level of resources consumption, the quota limit Q_{j_0} and the output prices, is estimated solving the linear programming problem shown in Eq. 4. The model follows the formulation originally proposed by Fare et al. [8], with an additional constraint to reflect the fact that the catches are bounded by the quotas (Q_{j_0}) imposed by fisheries regulatory conditions.

$$\max\left\{\sum_{r=1}^{s} p_{rj_0} y_r^0 \;\middle|\; \right.$$

$$\left.\sum_{j=1}^{n}\lambda_j y_{rj} \ge y_r^0, \quad r = 1,\ldots,s\right.$$

$$\begin{aligned}
&\sum_{r=1}^{s}\sum_{j=1}^{n}\lambda_j y_{rj} \le Q_{j_0}, \qquad (3)\\
&\sum_{j=1}^{n}\lambda_j x_{ij} \le x_{ij_0}, \quad i = 1,\ldots,m\\
&\lambda_j \ge 0, \quad j = 1,\ldots,n\\
&y_r^0 \ge 0, \quad i = 1,\ldots,m\Big\}
\end{aligned}$$

In the formulation above, p_{rj_0} is the price of output r for the DMU_{j_0} under assessment and y_{0r} is a variable that, at the optimal solution, gives the amount of output r to be produced by DMU_{j_0} in order to maximise the revenue, subject to the technological restrictions imposed by the existing production possibility set. Revenue efficiency is then obtained, for each DMU_{j_0}, as the ratio of current revenue observed at DMU_{j_0} to the maximum revenue estimated by the optimal solution to model (4), as follows:

$$\textit{Revenue efficiency}_{j_0} = \frac{\sum_{r=1}^{s} p_{rj_0} y_{rj_0}}{\sum_{r=1}^{s} p_{rj_0} y_r^{0*}}. \qquad (4)$$

In the context of fisheries studies, the revenue efficiency of a vessel indicates by how much the current revenue of a vessel could be increased without requiring an increase in the level of resources used or in the quota limits, or changes in the prices paid for the species landed. The increase in revenue must be achieved either by a proportional increase in the quantities captured of each species (measured in kg), corresponding to the estimate of technical efficiency, and/or by a different composition of captures, corresponding to the estimate of output allocative efficiency, which involves an optimization in the selection of the target species taking into account their relative prices. The relation between revenue efficiency and its components, associated to technical efficiency and allocative efficiency is as follows:

$$\textit{Revenue efficiency}_{j_0} = \textit{Technical efficiency}_{j_0} \times \textit{Allocative efficiency}_{j_0}. \qquad (5)$$

As a result, in the DEA framework, the measure of output allocative efficiency can be obtained residually as the ratio of revenue efficiency, obtained from expression (4), and the output oriented technical efficiency measure, obtained from model (2). The definition of output allocative efficiency is as follows: output allocative efficiency captures the ability of the DMU to choose the best mix of outputs (i.e., the best combination of species captured) in order to maximise revenue, given their relative prices. DEA model was implemented with AIMMS® and the remaining statistical analysis was undertaken using SPSS®.

3 Dredge Fleets that Operates in the Portuguese West Coast of Mainland Portugal

Currently the artisanal dredge fleet that operates in the west coast of mainland Portugal comprises 36 vessels (11 and 25 vessels operating in the Northwest and Southwest fishing areas respectively) (Fig. 1). Dredge vessels in the Northwest area have an overall length ranging from 10 to 16 m, an engine power between 73 and 128 kW, a gross tonnage (GT) between 9 and 25 tons and a crew composed of five fishermen, whereas in the Southwest area dredge vessels have an overall length ranging from 9 to 14 m, an engine power between 46 and 97 kW, a gross tonnage (GT) between 6 and 15 tons and a crew composed of 4–5 fishermen. The bivalve dredge fishery in the Northwest area is monospecific (single species) targeting the surf clam (*Spisula solida*), whilst in the southwest area the fishery is multispecific, targeting four species, the surf clam, the smooth clam (*Callista chione*), the donax clam (*Donax spp.*) and the pod razor clam (*Ensis siliqua*).

Although the majority of the management measures are similar in both fishing areas (e.g. seasonal closure, the minimum landing sizes and the gear specifications), there are differences in terms of the quota allocated. The quotas are reviewed on an annual basis considering the result of the annual monitoring surveys carried out by

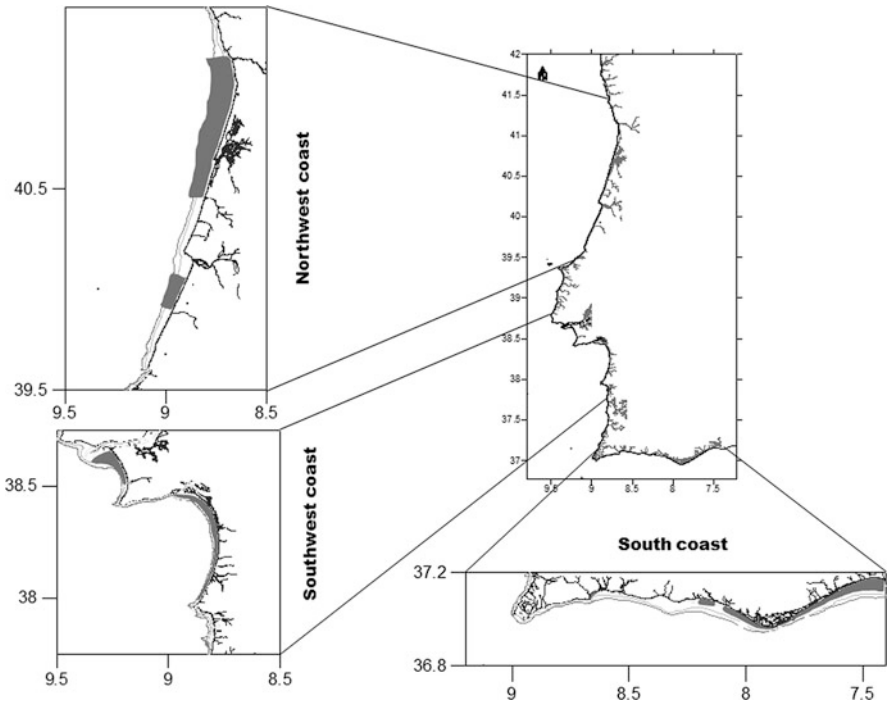


Fig. 1 Distribution of bivalve beds (*grey areas*) in the Northwest and Southwest fishing areas

the Portuguese Institute for the Ocean and Atmosphere (IPMA), and can be changed if necessary to adjust the catch to the status of the stocks (Oliveira et al. [21]).

4 Data

The dataset used in the present study was provided by the General Directorate of Natural Resources, Safety and Maritime Services (DGRM) and covers the period between January 2006 and December 2012. Of the dredge vessels that are currently licensed in the west coast (36 vessels) only 32 vessels were included in the analysis. The other 4 vessels (all from the Southwest dredge fleet) were excluded because in most of the years they used other fishing gears. Tables 1 and 2 present the average characteristics of the fleets that operated in the two areas, the mean fishing days per year, the biological stock indicator, the mean annual fishing quota per vessel, average yearly landings (in weight) and mean yearly price per kg at first sale. In the Southwest, price fluctuations do not occur (Table 2) since catches are sold through a contract that is established in the beginning of each year between the fishermen and the buyer. Therefore, although fishermen are obliged to pass the catches through the auction market, they are not obliged to sell them by auction. Thus, the selling price remains unchanged over the year. In the Northwest coast the price varies throughout the year because the species are sold at the auction (Table 1).

Concerning the variables defined for the DEA analysis, it was used one output for the Northwest area and four outputs for the Southwest areas, corresponding to total amount landed by species per year (Tables 1 and 2). To estimate the revenue efficiency of the fishing activity, it was collected data on the value of landings per species and vessel. For each area, the average price for each of the target species was calculated by dividing the landed value by the amount (kg) landed by each vessel. The input variables used were: overall vessel length (m), GT (ton), engine power of the vessel (kW), a biomass stock indicator for each of the target species (g per 5 min tow dredging) (fixed factors), and the effective days at sea (variable factor). The biomass stock indicator was obtained from bivalve research surveys conducted by IPMA, specifically designed to evaluate the conservation status of the commercial species. The surveys are carried out in a yearly basis onboard the IPMA research vessels. Details on both sampling design and procedures can be found in Rufino et al. [26]. The annual fishing quota per vessel (kg) was used as a contextualizing factor that bounds the output expansion allowed for each vessel and it was calculated by multiplying the daily/weekly quota for all target species and the effective days at sea/weeks per year for each vessel. The efficiency assessment models considered 73 DMUs for the Northwest fleet and 131 DMUs for the Southwest fleet. Since the potential differences in the operating conditions between the years are essentially related to stock abundance, and the models already incorporate the stock level as an input, the unit of analysis was the vessel's operation in a given year, run for a pooled sample with vessels from all years together.

Table 1 Profiling of the Northwest fleet (average values between 2006 and 2012)

	Northwest fleet (means)						
	2006	2007	2008	2009	2010	2011	2012
Inputs							
Overall length (meters)	13.3	13.3	13.3	13.3	13.4	13.4	13.4
GT (ton)	16.5	16.5	16.5	16.5	17.3	17.3	17.3
Engine power (kw)	103.3	103.3	103.3	103.3	103.2	103.2	103.2
No. days at sea	123.8	92.6	72.2	77.4	92.4	85.8	71.3
Surf clam stock (g per 5 min tow dredging)	33.4	16.9	11.1	28.5	45.9	54.3	55.6
Contextualizing factor							
Annual fishing quota per vessel kg)	62,400.0	62,400.0	62,400.0	62,400.0	62,400.0	72,000.0	72,000.0
Outputs							
Capture of surf clam (kg)	39,550.6	11,639.2	11,811.7	23,936.6	35,492.2	36,765.9	35,535.9
Output prices							
Prices of surf clam (€)	2.92	3.12	2.76	1.95	2.05	2.91	2.88

Table 2 Profiling of the Southwest fleet (average values between 2006 and 2012)

	Southwest fleet (means)						
	2006	2007	2008	2009	2010	2011	2012
Inputs							
Overall length (meters)	11.1	11.3	11.3	11.4	11.4	11.4	11.3
GT (ton)	9.5	9.7	9.7	10.0	10.0	10.0	9.7
Engine power (kw)	71.7	71.8	71.8	73.0	73.0	73.0	71.8
No. days at sea	147.6	144.2	144.1	149.1	89.2	77.6	122.3
Surf clam stock (g per 5 min tow dredging)	43.7	36.5	100.3	81.7	39.5	34.2	36.1
Smooth clam stock (g per 5 min tow dredging)	386.1	171.6	182.4	325.5	195.4	214.2	232.1
Donax clam stock (g per 5 min tow dredging)	29.5	81.5	100.9	107.5	63.0	102.3	110.6
Razor clam stock (g per 5 min tow dredging)	108.3	49.1	70.2	127.4	143.5	131.0	140.2
Contextualizing factor							
Annual fishing quota per vessel (kg)	88,533.3	86,490.0	86,430.0	99,000.0	99,000.0	99,000.0	99,000.0
Outputs							
Capture of surf clam (kg)	1375.0	1533.8	661.7	1195.9	585.7	196.7	1762.3
Capture of smooth clam (kg)	14,723.6	11,131.0	9288.5	9501.6	12,339.7	15,555.0	14,579.6
Capture of donax clam (kg)	2185.3	3040.4	5541.2	3166.4	1573.6	2057.0	4352.6
Capture of razor clam (kg)	8297.5	6064.6	4118.8	3415.4	2598.4	1067.4	2379.2
Output prices							
Prices of surf clam (€)	1.50						
Prices of smooth clam (€)	1.00						
Prices of donax clam (€)	2.50						
Prices of razor clam (€)	2.50						

5 Results and Discussion

5.1 Catch Composition

Figures 2 and 3 show the evolution of total landings during the period studied, for dredge vessels operating in the Northwest and Southwest areas, respectively. Since in the Northwest coast the fishery is monospecific (single species) the total amount landed reflects the conservation status of the *Spisula solida* stock (Table 1). In the case of the Southwest area it can be observed from Fig. 3 that although the catch composition varied among years, the total amount landed only changed slightly over the years. Since no significant changes in quotas occurred in the years studied, we believe that the changes observed in catch composition are only related to changes in the biomass stock or changes in demand of the bivalve market.

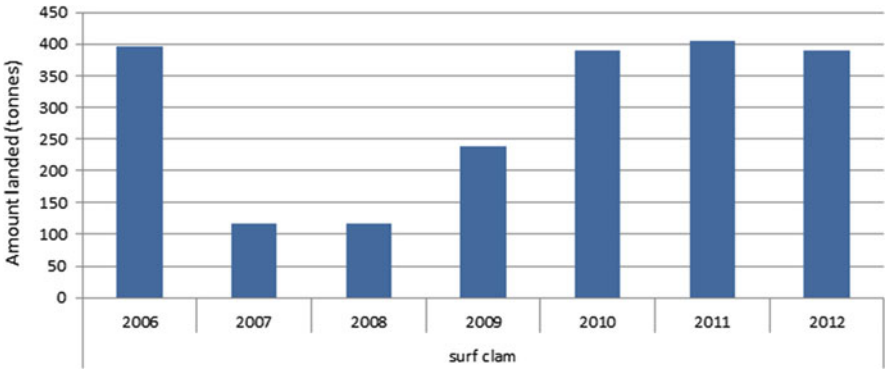


Fig. 2 Northwest fleet. Total landed from dredge vessels between 2006 and 2012

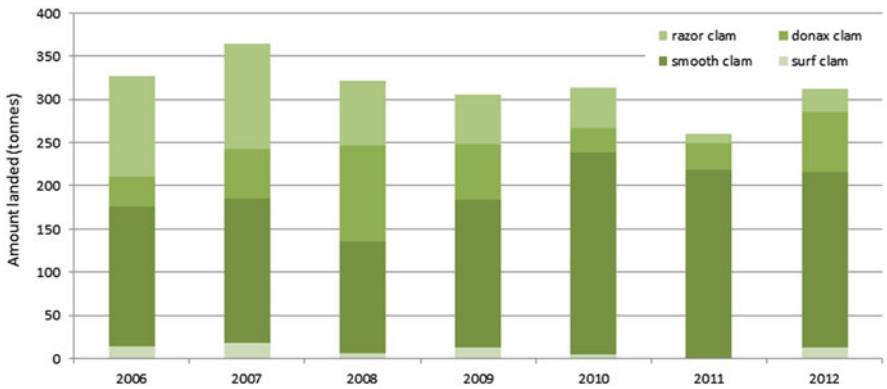


Fig. 3 Southwest fleet. Landed catch composition from dredge vessels between 2006 and 2012

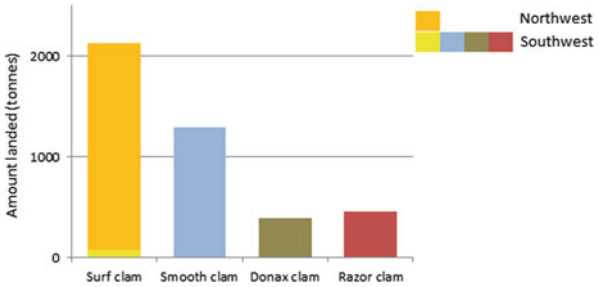


Fig. 4 Comparison of total landings per species in weight (tonnes) from Northwest and Southwest dredge vessels between 2006 and 2012

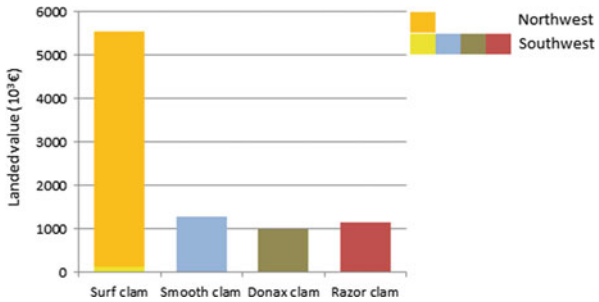


Fig. 5 Comparison of total landings per species in values (10^3€) from Northwest and Southwest dredge vessels between 2006 and 2012

Figures 4 and 5 show the total catch landed, in weight (tonnes) and in value (10^3€), respectively, for the species caught by the Northwest and Southwest dredge fleets. Despite harvesting only one species (surf clam) and having only half of the vessels of the Southwest fleet, it is important to highlight that the Northwest fleet achieved, on average, about 48 % of the total catch landed in both areas (in tonnes, see Fig. 4) and 60 % of the total income (Fig. 5) in this particular period of time (i.e., between 2006 and 2012). This could be related to the fleet ownership profile. In fact, in the Portuguese artisanal dredge fishery the skipper is usually the ship-owner. However, in the Southwest area, the ship-owners usually have several vessels and therefore can manage the activity of their vessels according to the oceanographic conditions and market demand. Indeed, the ship-owner can decide when and which vessels can fish to accomplish the order lists by species received on land before they sail.

5.2 Technical and Revenue Efficiency Analysis

Tables 3 and 4 summarise the efficiency results (geometric means, standard deviation and the number of efficient DMUs) for the Northwest and Southwest fleets between 2006 and 2012, respectively. The allocative efficiency is not presented for the Northwest fleet because as this fleet only harvests one species, the weight and value of the catches are proportional, and thus technical and revenue efficiency results are identical (Herrero and Pascoe [11]).

Table 3 Efficiency results for the Northwest fleet

	Technical efficiency			Revenue efficiency		
	Geometric mean	St. dev	No. eff. DMUs	Geometric mean	St. dev	No. eff. DMUs
2006	0.944	0.073	3	0.944	0.073	3
2007	0.498	0.168	0	0.498	0.168	0
2008	0.761	0.207	2	0.761	0.207	2
2009	0.717	0.148	0	0.717	0.148	0
2010	0.699	0.242	2	0.699	0.242	2
2011	0.782	0.171	0	0.782	0.171	0
2012	0.830	0.169	3	0.830	0.169	3

Concerning revenue efficiency, it was observed that Northwest vessels have an average efficiency score significantly higher (K-W, $p = 0.001$) than the Southwest vessels (0.735 and 0.631, respectively). Is important to underline that the efficiency scores of each fleet were calculated with reference to fleet-specific frontiers, as the sample of Northwest fleet vessels and Southwest fleet vessels were analysed separately. Therefore it cannot be concluded that higher efficiency scores represent better performance levels. Indeed, this result only indicates that the performance of Northwest vessels is more homogeneous than that of Southwest vessels, as the average distance to the efficient frontier is smaller.

In the Southwest fleet, it can be concluded that most inefficiencies are due to technical causes. The comparison of the technical and the allocative efficiency levels obtained through the 7 years, showed that in general the composition of the catches was good but the volume of the catches could have been improved, especially since 2009. In an attempt to explore whether scale inefficiency has a significant impact on artisanal dredge fisheries in these areas, it was undertaken a hypothesis test firstly proposed by Banker [2] for determining the type of returns to scale of the DMUs' activity. If the efficiency distributions obtained using the CRS and variable returns to scale (VRS) models (Banker et al. [3]) were similar, it would mean a scale inefficiency almost nonexistent, and thus there would not enough evidence to support the hypothesis that the DMUs' activity exhibited VRS. In these cases, the differences in the shape of the production frontier using CRS and VRS models may be due to random variations and not to the intrinsic VRS properties of DMUs' activities. The existence of VRS in vessels' activities was formally tested using the K-W test. For both fleets, the null hypothesis was rejected ($p = 0.044$ and $p = 0.000$ for Northwest and Southwest fleets, respectively) which indicates that the vessels are likely to operate under variable returns to scale, emphasizing that an increase in the resources does not always cause a proportional increase in the catches. The decomposition of technical efficiency (CRS estimate) into pure technical efficiency (VRS estimate) and scale efficiency components for both fleets is shown in Tables 5 and 6.

For the Northwest fleet, the best practice vessels in terms of revenue efficiency in the 7 year period were found to be slightly smaller than the fleet's average (with

Table 4 Efficiency results for the Southwest fleet

	Technical efficiency			Revenue efficiency			Allocative efficiency		
	Geometric mean	St. dev	No. eff. DMUs	Geometric mean	St. dev	No. eff. DMUs	Geometric mean	St. dev	No. eff. DMUs
2006	0.886	0.124	7	0.764	0.154	2	0.862	0.121	2
2007	0.878	0.170	10	0.766	0.208	6	0.872	0.118	6
2008	0.834	0.134	6	0.702	0.130	0	0.841	0.108	0
2009	0.615	0.194	2	0.510	0.141	1	0.829	0.132	1
2010	0.711	0.173	4	0.621	0.197	3	0.873	0.110	3
2011	0.620	0.210	2	0.530	0.235	2	0.854	0.131	2
2012	0.725	0.245	7	0.577	0.227	1	0.797	0.142	1

Table 5 VRS and scale efficiency for the Northwest fleet

	VRS efficiency			Scale efficiency		
	Geometric mean	St. dev	No. eff. DMUs	Geometric mean	St. dev	No. eff. DMUs
2006	0.950	0.065	3	0.994	0.011	4
2007	0.598	0.177	1	0.832	0.226	1
2008	0.956	0.068	7	0.796	0.211	2
2009	0.791	0.133	0	0.907	0.047	0
2010	0.702	0.246	3	0.995	0.011	6
2011	0.821	0.168	1	0.952	0.121	8
2012	0.892	0.113	4	0.930	0.163	9

Table 6 VRS and scale efficiency for the Southwest fleet

	VRS efficiency			Scale efficiency		
	Geometric mean	St. dev	No. eff. DMUs	Geometric mean	St. dev	No. eff. DMUs
2006	0.964	0.087	13	0.919	0.099	8
2007	0.951	0.095	14	0.923	0.127	11
2008	0.913	0.101	9	0.914	0.101	6
2009	0.655	0.203	2	0.939	0.074	2
2010	0.813	0.163	7	0.875	0.121	4
2011	0.802	0.195	6	0.773	0.194	2
2012	0.895	0.142	11	0.810	0.228	7

12.9 m length and an engine power of 96.3 kW), whereas in the Southwest fleet, the overall length and the engine power of the best practice vessels did not differ from the fleet's average (with 11.3 m length and an engine power of 71.8 kW). Statistically significant differences ($p = 0.005$) between the scale efficiency results of both fleets were observed confirming that scale efficiency is more prevalent in the Northwest fleet than in the Southwest fleet. A vessel is considered to be scale efficient when the combination of spent resources and volume of catches is optimal so that any modifications on this combination will result in efficiency loss. Thus the scale efficiency value is obtained by dividing the CRS efficiency by VRS efficiency. This means that despite the bivalve dredge fishery is monospecific in Northwest area, the fleet is technically close to the optimal operation.

Figures 6 and 7 compare the target landings (in tonnes) obtained as by-products of the technical efficiency and revenue efficiency models, respectively. The technical efficiency model suggests for each vessel a proportional increase in the amount landed for each species, whereas the revenue model allows changes to the mix of species captured. For the Northwest fleet both technical and revenue efficiencies suggest the same increment in each year, which is explained by the fact that this fleet only harvest one species (Fig. 6). The higher increments are required for the years 2007, 2010 and 2011 (99, 120 and 90 tonnes, respectively). These increments were coincident with the first year in which the biological stock indicator fell and the years of its recover.

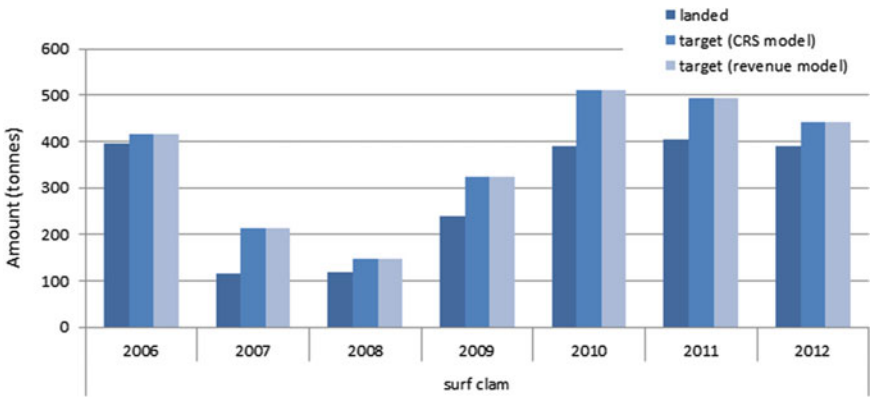


Fig. 6 Amount landed versus target landings for Northwest fleet

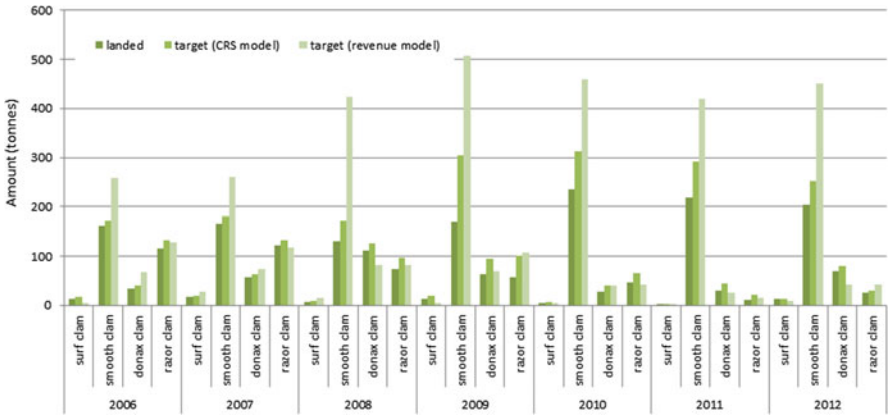


Fig. 7 Amount landed versus target landings for Southwest fleet

In the Southwest fleet, the species in which is required more often an increment in a revenue maximisation perspective is by far the smooth clam (94, 294, 337, 226 and 246 tonnes in 2007, 2008, 2009, 2010 and 2012, respectively). For the donax clam was also required an increment of 33 tonnes in 2006. A better strategy to maximize revenue also involves harvesting less quantities of three species, namely surf clam (9 and 8 tonnes, in 2006 and 2009, respectively), razor clam (4 tonnes in 2007 and 2010) and donax clam (27, 6 and 26 tonnes in 2008, 2011 and 2012). From a technical perspective a different scenario is presented. No reductions to catches are suggested, and the higher increments needed correspond to the smooth clam (133 and 48 tonnes in 2009 and 2012, respectively).

5.3 A Strategic Approach with DEA

The performance analysis of an organisation based on a portfolio of business, corresponding to different dimensions represented in a matrix, dates back to the 1960s. This technique, known as the growth-share matrix, was developed by the Boston Consulting Group (BCG) to support strategic options formulation. This technique was later adapted to the context of efficiency and profitability analysis by Boussofiane et al. [4]. The efficiency-profitability matrix is divided into 4 quadrants, where different profiles of units are likely to exist, although the precise boundary positions between quadrants are subjective. This approach was applied to the Southwest dredge fleet for the last 3 years of the study and the quadrants boundaries used were identical to those proposed by Oliveira et al. [20]. It is intended to explore the relationship between the technical-efficiency measure obtained from a DEA analysis and the allocative-efficiency measure, obtained as a by-product of the DEA analysis with a revenue maximization model.

Figure 8 illustrates the relationship between allocative efficiency and technical efficiency for the Southwest dredge vessels. The analysis of the allocative efficiency versus technical efficiency matrix is an alternative way to identify best-performing vessels, corresponding to those located in the top corner of each matrix that can be considered the "stars". There are nine vessels located in the "star vessels" quadrant at least once in the 3 years analysed. One of these vessels was fully efficient, both technically and allocatively, in 2 years (no. 8), and four vessels (no. 1, 6, 13 and 18) were only fully efficiency in one of the years. The other four vessels (no. 5, 10, 15 and 19) achieved high efficiency scores in all years, despite never achieving the maximum efficiency level.

The analysis of Fig. 8 also shows that every year there are five vessels located in the "problem vessels" quadrant. This suggests that there is scope for efficiency improvements in this fleet. Vessels located in the "problem vessels" quadrant have the potential for achieving greater technical and allocative efficiency levels, indicating that they should change the proportion among the species captured and, at the same time, they should increase the total amounts landed. Only one vessel is consistently located in this quadrant over the years (no. 2). In addition, three vessels (no. 4, 16 and 18) were classified as "problem vessels" in two of the years, so their activity should be carefully monitored, to identify the practices that need to be modified to improve their performance. The number of vessels in the "good capture composition" quadrant is higher compared with the number of vessels located in the "good in the amounts landed" quadrant. This suggests that fishermen behaviour focuses on capturing the species that can maximise revenue rather than only aiming at capturing large quantities of bivalves. Vessels located in the "good capture composition" quadrant need to focus on incrementing the amounts landed, keeping the current proportion among the target species harvested. Vessels located in the "good in the amounts landed" quadrant have an inappropriate choice of target species, and thus it may be possible to increase profits by redirecting captures towards other species. In certain cases, vessels previously referred as "star vessels"

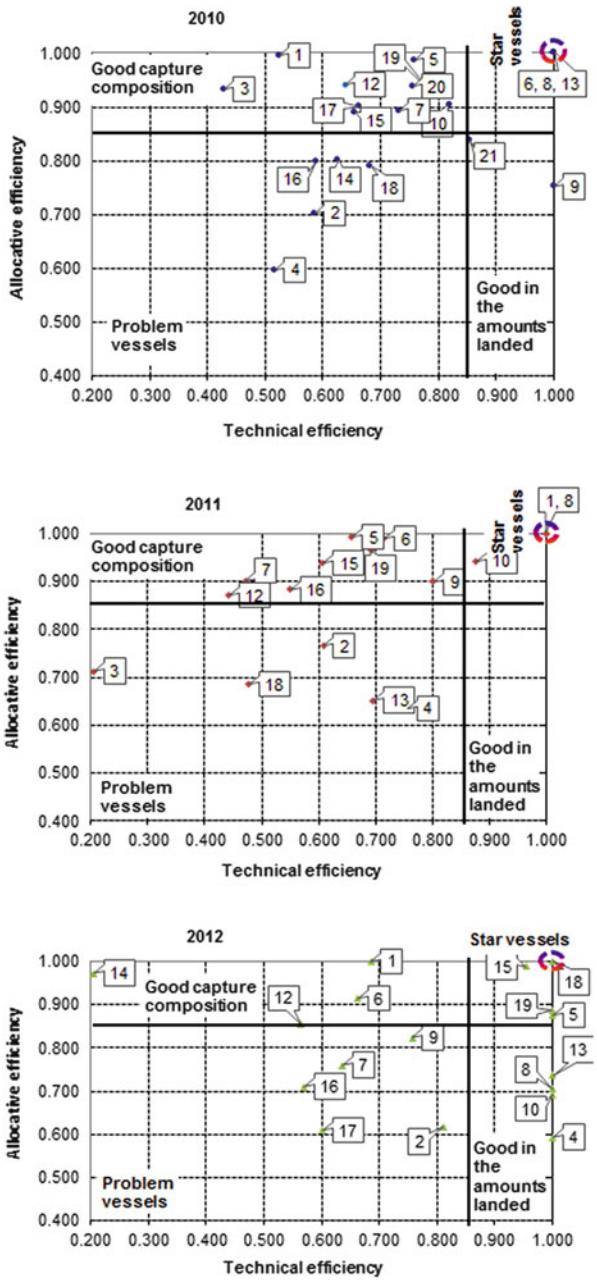


Fig. 8 Allocative efficiency versus technical efficiency matrix for the Southwest fleet

(e.g. no. 8 and 13) decreased their allocative or/and technical efficiency and fell in the "good in the amounts landed" quadrant.

6 Conclusions

The present study allowed clarifying same interesting issues about the performance of both Northwest and Southwest dredge fleets between 2006 and 2012. Concerning the composition of the catches, the amount landed in the Northwest area are directly related to stock since in this area the fishery is monospecific. In the Southwest the changes observed in catch composition are not only related to changes in the stock but also to changes in bivalve market demand since no significant changes in quotas occurred in the years considered. During the period studied, and despite harvesting only the surf clam species and having only half the vessels of the Southwest fleet, landings from the Northwest dredge fleet accounted for 48 % of the total catch landed in both areas, and 60 % of the total income. This result reflects the differences in the ownership profile. In contrast to the common in the artisanal dredge fishery where the skipper is usually the ship-owner, as it is the case of the dredge fleet that operates in the Northwest area, in the Southwest area, the ship-owners usually have several vessels, and manage their activity as a whole, according to the oceanographic conditions and market demand, instead of treating independently the different vessels. During periods of low demand, some vessels may remain inactive during several weeks decreasing, this way, their efficiency. This justifies the reason why the performance of the Northwest fleet is more homogeneous than the Southwest fleet. The analysis of returns to scale allowed concluding that both fleets are operating under variable returns to scale, meaning that in this fishery a possible increase in the resources could not imply a proportional increment in the catches landed. The BCG growth-share matrix constructed for the Southwest fleet allowed to explore graphically the relationship between allocative efficiency and technical efficiency for the last 3 years. The main management challenge concerns the vessels located in the "problem vessels" quadrant. They are not operating close to efficient levels neither in technical or allocative terms. In order to make the fishing activity more profitable, these vessels should change the balance between the species captured and the amounts landed. The vessels located in the "good capture composition" quadrant should increment the quantities landed in order to become "stars" and attain higher profits. The vessels located in the "good in the amounts landed" quadrant should redirect the fishing effort to capture a different mix of species. As they are close to operating efficiently in technical terms, the profitability can only be increased by changing the mix of species captured. Their activity should be redesigned in order to emulate the best-practices observed in the benchmark vessels of the same fleet. The present study emphasizes the importance of assessing efficiency in artisanal fishery. The results achieved allowed to better understanding fishing operation and how the fleets achieved their performance. In face of that and from a management perspective, the Northwest fleet should start diversifying the catch by targeting the

other bivalve species with commercial interest that occurs in the area in order to maximize their revenue, since in terms of the resources employed no changes are needed. Being restricted to a single species, the performance of the fleet is extremely dependent of the status of the stock.

Concerning the Southwest fleet, the improvement of the performance of the fleet is more difficult to achieve due to the ownership profile. Nevertheless, the results revealed, on general, that although the composition of catches is appropriate, the amount landed could be improved. Our suggestion would be to increase the catches of all species, perhaps directing the effort to those that have a higher market price. The results from the BCG growth-share matrix could also be useful if a vessels scrapping plan is put in place in this area aiming to adjust fishing effort to the status of the exploited stocks. Therefore, the vessels that should be scraped from the fishery should be those that are located in the "problem vessels" quadrant.

Acknowledgements The funding of this research through the scholarship SFRH/BD/77058/2011 from the Portuguese Foundation of Science and Technology (FCT) is gratefully acknowledged. This study was undertaken under the project framework "Desarrollo sostenible de las pesquerías artesanales del Arco Atlántico" (PRESPO) of the INTERREG IVB Programme-Atlantic Arc, co-funded by the EU (ERDF Programme).

References

1. Alam, M.F., Murshed-e-Jahan, K.: Resource allocation efficiency of the prawn-carp farmers of Bangladesh. Aquac. Econ. Manag. **12**, 188–206 (2008)
2. Banker, R.D.: Maximum likelihood, consistency and data envelopment analysis: a statistical foundation. Manag. Sci. **39**(10), 1265–1273 (1993)
3. Banker, R.D., Charnes, A., Cooper, W.W.: Some models for estimating technical and scale inefficiencies in data envelopment analysis. Manag. Sci. **30**, 1078–1092 (1984)
4. Boussofiane, A., Dyson, R.G., Thanassoulis, E.: Applied data envelopment analysis. Eur. J. Oper. Res. **52**(1), 1–15 (1991)
5. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring efficiency of decision making units. Eur. J. Oper. Res. **2**, 429–444 (1978)
6. Cooper, W.W., Seiford, L.M., Tone, K.: Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software. Kluwer, Boston (2000)
7. Dupont, D., Grafton, R., Kirkley, J., Squires, D.: Capacity utilization measures and excess capacity in multi-product privatized fisheries. Resour. Energy Econ. **24**(3), 193–210 (2002)
8. Fare, R., Grosskopf, S., Lovell, C.A.K.: The Measurement of Efficiency of Production. Kluwer-Nijhoff Publishing, Boston (1985)
9. Farrell, M.J.: The measurement of productive efficiency. J. R. Stat. Soc. Ser. A, Gen. **120**(Part 3), 253–281 (1957)
10. Guyader, O., Berthou, P., Koustikopoulos, C., Alban, F., Demanèche, S., Gaspar, M.B., Eschbaum, R., Fahy, E., Tully, O., Reynal, L., Curtil, O., Frangoudes, K., Maynou, F.: Small scale fisheries in Europe: a comparative analysis based on a selection of case studies. Fish. Res. **140**, 1–13 (2013)
11. Herrero, I., Pascoe, S.: Value versus volume in the catch of the Spanish South-Atlantic trawl fishery. J. Agric. Econ. **54**(2), 325–341 (2003)
12. Hoff, A.: Bootstrapping Malmquist indices for Danish seiners in the North Sea and Skagerrak. J. Appl. Stat. **33**(9), 891–907 (2006)

13. Idda, L., Madau, F.A., Pulina, P.: Capacity and economic efficiency in small-scale fisheries: evidence from the Mediterranean Sea. Mar. Policy **33**, 860–867 (2009)
14. Kirkley, J., Squires, D., Strand, I.: Assessing technical efficiency in commercial fisheries: the Mid-Atlantic sea scallop fishery. Am. J. Agric. Econ. **77**, 686–697 (1995)
15. Kirkley, J., Squires, D., Strand, I.: Characterizing managerial skill and technical efficiency in a fishery. J. Product. Anal. **9**, 145–160 (1998)
16. Kirkley, J., Squires, D., Alam, M., Ishak, H.: Excess capacity and asymmetric information in developing country fisheries: the Malaysian purse seine fishery. Am. J. Agric. Econ. **85**(3), 647–662 (2003)
17. Koopmans, T.C.: Activity Analysis of Production and Allocation. Willey, New York (1957)
18. Lindebo, E., Hoff, A., Vestergaard, N.: Revenue-based capacity utilisation measures and decomposition: the case of Danish North Sea trawlers. Eur. J. Oper. Res. **180**(1), 215–227 (2007)
19. Maravelias, C., Tsitsika, E.: Economic efficiency analysis and fleet capacity assessment in Mediterranean fisheries. Fish. Res. **93**, 85–91 (2008)
20. Oliveira, M.M., Camanho, A.S., Gaspar, M.B.: Technical and economic efficiency analysis of the Portuguese artisanal dredge fleet. ICES J. Mar. Sci. **67**(8), 1811–1821 (2010)
21. Oliveira, M.M., Camanho, A.S., Gaspar, M.B.: The influence of catch quotas on the productivity of the Portuguese bivalve dredge fleet. ICES J. Mar. Sci. **70**(7), 1378–1388 (2013)
22. Oliveira, M.M., Camanho, A.S., Gaspar, M.B.: Enhancing the performance of quota managed fisheries using seasonality information: the case of the Portuguese artisanal dredge fleet. Mar. Policy **45**(3), 114–120 (2014)
23. Pascoe, S., Coglan, L.: Contribution of unmeasurable factors to the efficiency of fishing vessels. Am. J. Agric. Econ. **84**, 585–597 (2002)
24. Pascoe, S., Herrero, I.: Estimation of a composite fish stock index using Data Envelopment Analysis. Fish. Res. **69**, 91–105 (2004)
25. Pascoe, S., Tingley, D.: Economic capacity estimation in fisheries: a non-parametric ray approach. Resour. Energy Econ. **28**, 124–138 (2006)
26. Rufino, M.M., Gaspar, M.B., Pereira, A.M., Maynou, F., Monteiro, C.C.: Ecology of megabenthic bivalve communities from sandy beaches on the south coast of Portugal. Sci. Mar. **74**(1), 163–178 (2010)
27. Sharma, K., Leung, P.: Technical efficiency of the longline fishery in Hawaii: an application of a stochastic production frontier. Mar. Resour. Econ. **13**, 259–274 (1999)
28. Sharma, K.R., Leung, P., Chen, H., Peterson, A.: Economic efficiency and optimum stocking densities in fish polyculture: an application of data envelopment analysis (DEA) to Chinese fish farms. Aquaculture **180**, 207–221 (1999)
29. Tingley, D., Pascoe, S.: Eliminating excess capacity: implications for the Scottish fishing industry. Mar. Resour. Econ. **20**, 407–424 (2005)
30. Tingley, D., Pascoe, S.: Factors affecting capacity utilization in English channel fisheries. J. Agric. Econ. **56**, 287–305 (2005)
31. Tingley, D., Pascoe, S., Mardle, S.: Estimating capacity utilisation in multi-purpose, multi-metier fisheries. Fish. Res. **63**(1), 121–134 (2003)
32. Tingley, D., Pascoe, S., Coglan, L.: Factors affecting technical efficiency in fisheries: stochastic production frontier versus data envelopment analysis approaches. Fish. Res. **73**(3), 363–376 (2005)
33. Vestergaard, N., Squires, D., Kirkley, J.: Measuring capacity and capacity utilization in fisheries: the case of the Danish gill-net fleet. Fish. Res. **60**(2–3), 357–368 (2003)

Production Planning of Perishable Food Products by Mixed-Integer Programming

Maria João Pires, Pedro Amorim, Sara Martins, and Bernardo Almada-Lobo

Abstract In this paper, the main complexities related to the modeling of production planning problems of food products are addressed. We start with a deterministic base model and build a road-map on how to incorporate key features of food production planning. The different "ingredients" are organized around the model components to be extended: constraints, objective functions and parameters. We cover issues such as expiry dates, customers' behavior, discarding costs, value of freshness and age-dependent demand. To understand the impact of these "ingredients", we solve an illustrative example with each corresponding model and analyze the changes on the solution structure of the production plan. The differences across the solutions show the importance of choosing a model suitable to the particular business setting, in order to accommodate the multiple challenges present in these industries. Moreover, acknowledging the perishable nature of the products and evaluating the amount and quality of information at hands may be crucial in lowering overall costs and achieving higher service levels. Afterwards, the deterministic base model is extended to deal with an uncertain demand parameter and risk management issues are discussed using a similar illustrative example. Results indicate the increased importance of risk-management in the production planning of perishable food goods.

1 Introduction

The supply chain planning of food products is ruled by the dynamic nature of its products. Throughout the planning horizon, the characteristics of these products go through significant changes. The root cause for these changes may be related to, for example, the physical nature of the products or the value that the customer lends to them. Without acknowledging the perishable nature of food products, one may incur in avoidable spoilage costs (for example, in the case of meat products) or, on the other hand, sell the product before it is close enough to its best state (for

M.J. Pires • P. Amorim (✉) • B. Almada-Lobo • S. Martins
INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
e-mail: maria.pires@fe.up.pt; amorim.pedro@fe.up.pt; almada.lobo@fe.up.pt; sara.martins@fe.up.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_19

example, in the case of cheese products). In this paper, we focus on perishable food products that start worsening their properties after being produced.

Fleischmann et al. [7] define planning as the activity that supports decision-making by identifying the potential alternatives and making the best decisions according to the objective of the planners. Let us look into the specific challenges of engaging in a production planning activity in the context of food products.

In order to identify the alternatives it is important to frame the decisions that the decision maker wants to make. It is common to organize the supply chain planning according to two dimensions: the supply chain process (procurement, production, distribution and sales) and the hierarchical level (strategic, tactical and operational). The scope of this paper is in the production supply chain process and we deal with problems arising at the tactical/operational decision level. Therefore, we will address food production planning problems that have to decide about the size of the lots to be produced and about the schedule of these production lots. In this problem, we usually determine the size of lots to be produced while trading off the changeover and stock holding costs. In food production, expiry dates may enforce constraints related to the upper bounds on lot-sizes and consequently the need of scheduling more often a given family of products (increasing the difficulty of sequencing). Expiry dates relate to the concept of perishability that is defined by Amorim et al. [4] as: "A good, which can be a raw material, an intermediate product or a final one, is called 'perishable' if during the considered planning period at least one of the following conditions takes place: (1) its physical status worsens noticeably (e.g. by spoilage, decay or depletion), and/or (2) its value decreases in the perception of a(n internal or external) customer, and/or (3) there is a danger of a future reduced functionality in some authority's opinion.". In this paper, we will consider goods that suffer a physical deterioration, for which customers' attribute a decreasing value and for which authorities usually limit the commercialization period.

The second part of Fleischmann et al. [7] definition of planning relates to the objectives of the planners. The literature in production planning tackles most of the problems with traditional single objective models. The goal is usually related either to an operational measure, such as makespan, or to some monetary measure, such as cost or profit. In this paper, we show the interest of extending these objectives by including factors related to the food industry, such as spoilage costs. Moreover, the use of a multi-objective approach is described in order to account for the customer willingness for fresher products and to induce a risk conscious strategy. Acknowledging freshness in production planning besides avoiding products' spoilage, may yield a substantial intangible gain derived from delivering fresher products to customers. Such considerations are closely related to the consumer purchasing behavior of perishable goods that should be the concern of any planner in a (food) company with a market orientation.

The key contribution of this paper is to provide a systematic approach to a problem that has been tackled sparsely in the literature. We believe that this road-map on mixed-integer models for production planning of perishable food products may be useful to any researcher or practitioner willing to start solving a problem in this field. For an extensive review in production planning problems dealing

with perishability the readers are referred to Pahl and Voß [9] and for a more comprehensive review on supply chain planning problems dealing with perishability the readers are referred to Amorim et al. [4].

In the remainder of the paper, we present how a traditional base model dealing with the production planning of food products has to be changed in order to accommodate the characteristics of the products it has to deal with. Therefore, we start by presenting a deterministic base model in Sect. 2. In Sect. 3, we understand how the constraints have to be extended to incorporate key aspects, such as the fact that products have a limited shelf-life or that customers pick up the fresher available products. Section 4 analyses the possible changes in the objective function: discarding costs of perished goods and valuing in a different objective function the freshness. Section 5 tackles the possibility of having more information on key parameters – dependency between price and age and between demand and age. The "ingredients" presented throughout these section can be mixed together in various ways to form the "recipe" suitable for the production environment. In order to help understanding the implications of these "ingredients" in the solution structure, all models are solved for an illustrative example in Sect. 6. Section 7 discusses the extension of the deterministic base model to a stochastic setting in which demand or other parameters may be uncertain leading the notion of a risk-conscious planning. This model road-map is summarized in Fig. 1. Finally, in Sect. 8 the main conclusions are presented.

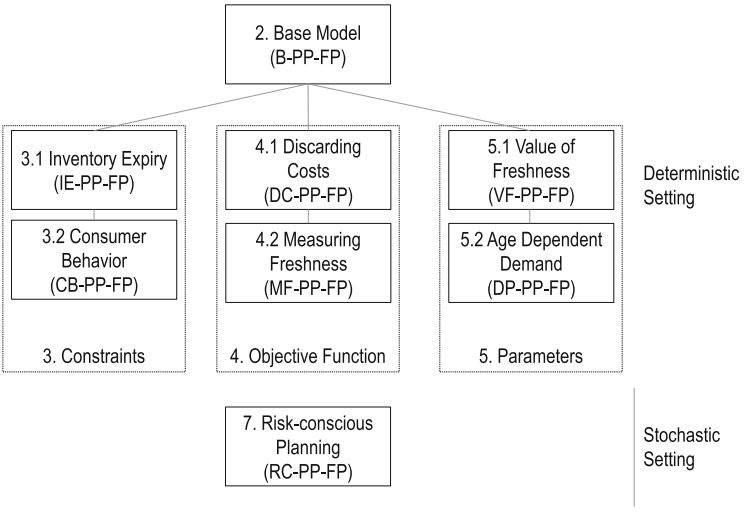


Fig. 1 Road-map of the different "ingredients" presented in this paper

2 Base Model

We start by presenting a base model for production planning in food industries. This model focuses on the packaging stage and has no considerations about the perishable nature of the products.

One important concept in the fast moving consumer food goods is the recipe. Usually, products belong to a certain recipe that requires a major setup and the products within the recipes just need a minor setup. This is known as production wheel policy by practitioners. We use an adaptation of the block planning formulation [8] that was designed for similar production environments to that of the food industry. To make it clearer, a block corresponds to a recipe and within and between recipes the sequence of products is set a priori. Therefore, the only decision to be made for each block/product, besides the sizing of the lots, is whether to produce it or not. This modeling approach increases the application potential of decision support systems in production planning, because decision makers are comfortable with the definition of the recipes and, simultaneously, the scheduling complexity is fairly reduced increasing the computational tractability of the related problems. In Fig. 2 a production schedule with two blocks, A and B, is depicted. Notice that before producing products of a given recipe a major setup is necessary. Afterwards, all products within the same recipe are produced after doing a minor setup. Block A has usually a lighter color or a less intense flavor than block B. Examples of this recipe structure can be found in the yoghurt, milk, juice and chocolate industries.

Let us now move to a formal description of the problem. Consider a set of products $k = 1, \ldots, K$ that are produced based on a certain recipe/block $j = 1, \ldots, N$. There is only one recipe to produce each product and, therefore, a product is assigned to one block only. Hence, for each block j there is a set $\mathscr{K}_j$ of products k related to it. Blocks are to be scheduled on $l = 1, \ldots, L$ parallel production lines over a finite planning horizon consisting of periods $t = 1, \ldots, T$ with a given length. This length is related to the company's practice of measuring external elements, such as demand or perishability (thus, periods correspond to days, weeks or months in most of the tactical/operational cases). According to the block structure, all scheduling decisions are already made for both recipes and products. Hence, the production sequence is determined beforehand, minimizing the setup times and costs according to the planner expertise [8]. This is particularly useful in practice, since companies have difficulties in measuring setups costs and setups times accurately. This limitation may reduce the applicability of traditional production planning objective functions.

Consider the following indices, parameters, and decision variables that are used hereafter.

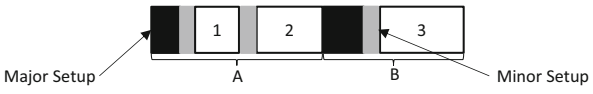


Fig. 2 Adapted block planning concept [5]

Indices

$l \in \mathscr{L}$	parallel production lines
$j \in \mathscr{N}$	blocks
$k \in \mathscr{K}_{\mathscr{J}}$	products
$t \in \mathscr{T}$	periods
$a \in \mathscr{A} = \{a \in \mathbb{Z}_0^+, t \in \mathscr{T} \mid a \leq t-1\}$	age (in periods)

Parameters

C_{lt}	capacity (time) of production line l available in period t
e_{lk}	capacity consumption (time) needed to produce one unit of product k on line l
c_{lk}	production costs of product k (per unit) on line l
u_k	shelf-life of product k just after production (time)
p_k	price of product k
h_k	inventory carrying cost of product k
m_{lj}	minimum lot size (units) of block j on line l
$\bar{s}_{lj}(\bar{\tau}_{lj})$	setup cost (time) of a changeover to block j on line l
$\underline{s}_{lk}(\underline{\tau}_{lk})$	setup cost (time) of a changeover to product k on line l
d_{kt}	demand for product k in period t (units)

Decision Variables

$\rho_{kt}^a \geq 0$	initial inventory of product k with age a available at period t
$\psi_{kt}^a \geq 0$	fraction of the maximum demand for product k delivered with age a at period t
$q_{lkt} \geq 0$	quantity of product k produced in period t on line l
$p_{lkt} \in \{0, 1\}$	equals 1, if line l is set up for product k in period t (0 otherwise)
$y_{ljt} \in \{0, 1\}$	equals 1, if line l is set up for block j in period t (0 otherwise)

From the decision variables it is noticeable that we use an adaptation of the simple plant location (SPL) reformulation to model inventory and demand fulfillment decision variables. In the traditional SPL reformulation [6], it is known for which period the production of a given period refers to. In a food production planning context we are more interested in tracing the actual age of the product. Therefore, in this case, we know for each period the age of the inventory of a given product. This will be rather helpful in limiting the usage of stock based on the shelf-life of the products. Moreover, it can also be used to keep track of the freshness of the products delivered to the clients. These potentialities will be further explored in the next sections. Figure 3 shows how traditional decision variables for production quantities (q_{lkt}) are transformed through the adapted SPL reformulation. Basically,

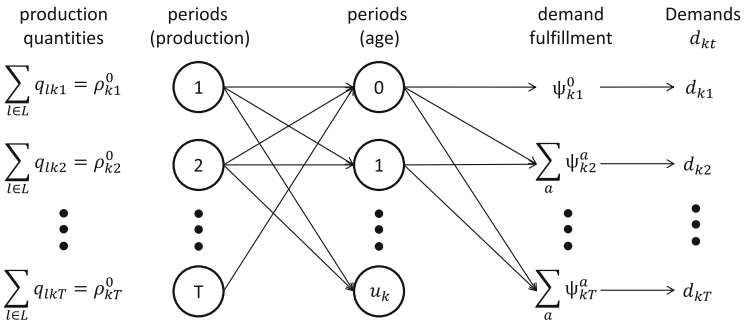


Fig. 3 Schematic representation of the adaptation of the simple plant location reformulation with an emphasis on the inventory age

the products produced in a given period correspond to the inventory with age 0 (ρ^0_{kt}). This inventory has its age updated throughout the planning horizon and it has a straight correspondence to the age of the products when fulfilling demand (ψ^a_{kt}).

The deployment of these two adapted concepts (block planning and simple plant location) results in a base model flexible enough to cope with the basic exigencies of production planning in food industries.

The base production planning model of food products (B-PP-FP) reads:

B-PP-FP

$$\max \sum_{k,t,a} p_k\, d_{kt}\, \psi^a_{kt} - \sum_{l,j,t} \bar{s}_{lj}\, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk}\, p_{lkt} + c_{lk}\, q_{lkt}) - \sum_{k,t,a} h_k\, (\rho^a_{kt} - d_{kt}\psi^a_{kt}) \quad (1)$$

subject to:

$$\sum_a \psi^a_{kt} \leq 1 \quad \forall k \in \mathscr{K}, t \in \mathscr{T} \quad (2)$$

$$\rho^a_{kt} = \rho^{a-1}_{k,t-1} - d_{k,t-1}\psi^{a-1}_{k,t-1} \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} \setminus \{0\} \quad (3)$$

$$\sum_l q_{lkt} = \rho^0_{kt} \quad \forall k \in \mathscr{K}, t \in \mathscr{T} \quad (4)$$

$$p_{lkt} \leq y_{ljt} \quad \forall l \in \mathscr{L}, j \in \mathscr{N}, k \in \mathscr{K}_{\mathscr{J}}, t \in \mathscr{T} \quad (5)$$

$$q_{lkt} \leq \frac{C_{lt}}{e_{lk}} p_{lkt} \quad \forall l \in \mathscr{L}, k \in \mathscr{K}, t \in \mathscr{T} \quad (6)$$

$$\sum_j \bar{\tau}_{lj}\, y_{ljt} + \sum_k (\underline{\tau}_{lk}\, p_{lkt} + e_{lk}\, q_{lkt}) \leq C_{lt} \quad \forall l \in \mathscr{L}, t \in \mathscr{T} \quad (7)$$

$$\sum_{k \in \mathscr{K}_{\mathscr{J}}} q_{lkt} \geq m_{lj} y_{ljt} \quad \forall l \in \mathscr{L}, j \in \mathscr{N}, t \in \mathscr{T} \tag{8}$$

$$\psi^a_{kt},\ \rho^a_{kt},\ q_{lkt} \geq 0;\ p_{lkt},\ y_{ljt} \in \{0,1\} \tag{9}$$

The objective function (1) maximizes the profit of the producer over the planning horizon. Therefore, revenue that comes from sold products is subtracted by setup costs of recipes, setup costs of products, variable production costs and inventory costs. Note that the setup structure considers major and minor setup for the first product to be produced in a given block. For example, in the yoghurt production when changing from one kind of yoghurt to another a major setup might correspond to cleansing the lines and linking the new yoghurt tank, while the minor setup may correspond to setting up the machine to fill the yoghurt in a different type of package. These two operations can seldom be done in parallel.

Constraints (2) forbid the sum of all sold products of different ages to exceed the demand. Constraints (3) establish the inventory balance constraints, ageing the stock throughout the horizon. They state that the inventory of a given age is equal to the inventory in previous period with a younger age subtracted by the amount of products that was sold with the same younger age. Constraints (4) link the production variables to the inventory ones, setting all production in a given period in all lines to the initial stock with age 0. Constraints (5) and (6) ensure that a product can only be produced if both the correspondent block and product are set up, respectively. Limited capacity in the lines is to be reduced by setup times between blocks, setup times between products and also by the time consumed producing products (7). Constraints (8) introduce minimum lot-sizes for each block.

Final constraints (9) define the domain of the decision variables.

3 Extending the Constraints of the Base Model

Two main realistic factors may impact the production plans of perishable food products: the fact that inventory that is beyond the expiry date can no longer be sold (product-related), and the fact that customers in face of inventories with different shelf-lives, choose products with the farthest expiry date (customer-related). These issues are addressed in turn by limiting the feasibility domain as follows.

3.1 Inventory Expiry Constraints

In order to make sure that no expired products are used to satisfy demand it suffices to redefine the demand fulfillment related constraints dealing with these variables.

The production planning model of food products with inventory expiry constraints (IE-PP-FP) reads:

IE-PP-FP

$$\max \sum_{k,t,a} p_k \, d_{kt} \, \psi_{kt}^a - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) - \sum_{k,t,a} h_k \, (\rho_{kt}^a - d_{kt} \psi_{kt}^a)$$

subject to:

$$\sum_{a \leq u_k - 1} \psi_{kt}^a \leq 1 \quad \forall k \in \mathscr{K}, t \in \mathscr{T} \tag{10}$$

$$\rho_{kt}^a = \rho_{k,t-1}^{a-1} - d_{k,t-1} \psi_{k,t-1}^{a-1} \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} \setminus \{0\} : a \leq u_k \tag{11}$$

(4), (5), (6), (7), and (8)

$$\psi_{kt}^a, \; \rho_{kt}^a, \; q_{lkt} \geq 0; \; p_{lkt}, \; y_{ljt} \in \{0, 1\}$$

In constraint (10) we now limit the age of the products used to fulfill demand to be strictly below the product's shelf-life (u_k). Constraint (11) updated the age of the products in stock until products reach their respective shelf-life. In fact, the market conditions can be even more adverse. Retailers usually do not accept products that have already passed one third of their total shelf-life. The remaining constraints are exactly the same as in the base model of Sect. 2.

3.2 Consumer Behaviour Constraints

In a context where the production process is tightened to the downstream supply chain processes satisfying final customers demand, it may be important to better incorporate the instinctive behaviour of consumers. Regarding food products, usually a last-expired-first-out (LEFO) policy is put in practice by customers. This behaviour may guide production plans towards a more just-in-time philosophy in which products' freshness is a priority.

It is necessary to add a new decision variable θ_{kt}^a in order to model this behaviour that equals 1, if inventory of product k with age a is used to satisfy demand in period t (0 otherwise).

The production planning model of food products incorporating consumer behaviour (CB-PP-FP) reads:

CB-PP-FP

$$\max \sum_{k,t,a} p_k \, d_{kt} \, \psi_{kt}^a - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) - \sum_{k,t,a} h_k \, (\rho_{kt}^a - d_{kt} \psi_{kt}^a)$$

subject to:

$$\psi_{kt}^a \leq \theta_{kt}^a \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} : a \leq u_k - 1 \tag{12}$$

$$\rho_{kt}^{a-1} - d_{kt} \psi_{kt}^{a-1} \leq M(1 - \theta_{kt}^a) \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} \setminus \{0\} : a \leq u_k - 1 \tag{13}$$

$$(10), (11), (4)(5), (6), (7), \text{and } (8)$$

$$\psi_{kt}^a, \, \rho_{kt}^a, \, q_{lkt} \geq 0; \, \theta_{kt}^a, \, p_{lkt}, \, y_{ljt} \in \{0, 1\} \tag{14}$$

In the previous models, it is assumed that the seller is able to assign optimal inventory quantities of different ages to customers in order to maximize profit. With constraints (12) and (13) this advantage no longer holds as the more instinctive consumer purchasing behaviour of perishable products that will drive customers to pick up products with the highest degree of freshness is mimicked. Thus, constraints (12) turn the value of θ_{kt}^a to 1, whenever inventory of a given product k in period t with age a is used to satisfy demand. The value of this variable θ_{kt}^a is used in constraints (13) to ensure that a fresher inventory can only be used after depleting the older inventory. In these constraints every time inventory of age a from a product k in period t is used (θ_{kt}^a), then either all fresher inventory was used to satisfy demand ($\rho_{kt}^{a-1} - d_{kt} \psi_{kt}^{a-1} = 0$) or there was no such younger inventory ($\rho_{kt}^{a-1} = 0$). Note that parameter M denotes a big number.

4 Extending the Objective Function of the Base Model

The most common approach to grasp the perishability phenomena is to penalize the spoiled products with a discard cost in the objective function. This penalty cost makes sense if we acknowledge that products have a limited shelf-life and probably an associated discarding cost. Another approach derives from the awareness of the customers' willingness to pay for fresher products while, simultaneously, the level of information regarding the detailed values of this willingness to pay is low. In this case, a new objective function is added to the one maximizing profit, aiming at maximizing the freshness of the products delivered.

4.1 Discarding Costs in the Objective Function

By incorporating discarding costs we extend the traditional production planning objective function by incorporating perishability related costs. We define the cost of spoiled products ($\bar{p}_k$) as an opportunity cost. This opportunity cost corresponds to the revenue yielded by the best alternative that could have been produced and sold instead of producing product k that got spoiled. However, it may also be regarded, in a more tangible manner, as a disposal cost for each unit of perished inventory that has to be properly discarded.

The production planning model of food products including discarding costs (DC-PP-FP) reads:

DC-PP-FP

$$\max \sum_{k,t,a} p_k \, d_{kt} \, \psi^a_{kt} - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) - \sum_{k,t,a} h_k \, (\rho^a_{kt} - d_{kt} \psi^a_{kt}) - \sum_{k,t,a \geq u_k} \bar{p}_k \rho^a_{kt} \qquad (15)$$

subject to:

$$(2), (3), (4), (5), (6), (7), \text{and } (8)$$

$$\psi^a_{kt}, \; \rho^a_{kt}, \; q_{lkt} \geq 0; \; p_{lkt}, \; y_{ljt} \in \{0, 1\}$$

The only difference to the model presented in Sect. 2 is reflected by the cost of spoilage tracked by the last term of (15). This cost is incurred whenever we hold stock that is beyond the product's shelf-life ($\rho^a_{kt} > 0 : a \geq u_k$).

4.2 Measuring Freshness as an Objective Function

In this model, the economic tangible profit is separated from the customer intangible value of having fresher products available in two distinct objective functions. The main motivation for such splitting comes from the fact that finding the willingness to pay for different customers is rather difficulty and lengthy to grasp in practice. The first objective continues to be the maximization of profit and the second one maximizes the average freshness of delivered products [2]. These two objectives are certainly conflicting since achieving a higher freshness of products delivered has to be done at the expense of more production lots that lead to higher setup costs. Therefore, we acknowledge the complete different nature of the two complementary

objectives and the difficulty to attribute different monetary values to different degrees of freshness. As a result, the decision maker/planner will be offered a trade-off between freshness of delivered products and total profit. This trade-off can be represented by a set of solutions which do not dominate one another regarding both objectives (non-dominated or Pareto optimal front). We need to define the following additional parameter $[d_{kt}]$ that is the number of non-zero occurrences in the demand matrix. This parameter is useful to have a more straightforward interpretation of the objective function value.

The model that accounts for a measure of freshness (MF-PP-FP) reads:

MF-PP-FP

$$\max \sum_{k,t,a} p_k \, d_{kt} \, \psi_{kt}^a - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) \tag{16}$$

$$\max \frac{1}{[d_{kt}]} \sum_{k,t,a} \frac{u_k - a}{u_k} \psi_{kt}^a \tag{17}$$

subject to:

$$(2), (3), (4), (5), (6), (7), \text{and } (8)$$

$$\psi_{kt}^a, \; \rho_{kt}^a, \; q_{lkt} \geq 0; \; p_{lkt}, \; y_{ljt} \in \{0, 1\}$$

The first objective function (16) maximizes profit in a similarly way of the base model. In the second objective (17) the mean freshness of products to be delivered is maximized. The number of periods before spoilage is estimated by $u_k - a$ and it is then normalized by the estimated shelf-life of the corresponding product. The cardinality of the non-zero demand occurrences is used to normalize this objective function between 0 and 1. This cardinality, for a given input set data, is constant and easily computed. Therefore, a value of 1 means that all products are delivered to customers in their fresher state.

This approach for modelling the production planning for food products has an interesting aspect to consider regarding inventory costs. When maximizing freshness in the second objective we are already trying to minimize stocks since we try to produce as late as possible in order to deliver products that were just produced. Therefore, if we had also included inventory costs in the first objective we would be somehow duplicating the inventory carrying cost effect and objective functions (16) and (17) would be too correlated (which must be avoided in multi-objective optimization).

5 Extending the Parameters of the Base Model

Another form of differentiating the base model of food production planning is by changing or detailing the input parameters, namely: price and demand. The key reasoning is that with more accurate information and more transparency across the supply chain partners, it would be possible to discriminate either price or demand in function of the actual age of the products.

5.1 Value of Freshness Parameter

In this model it is assumed that either the retailer or the final customer will be willing to pay a different price for products with different standards of freshness. Therefore, the price parameter is extended to $\hat{p}_k^a$, price of product k paid when the product has an age a.

The production planning model of food products with different freshness values (VF-PP-FP) reads:

VF-PP-FP

$$\max \sum_{k,t,a} \hat{p}_k^a \, d_{kt} \, \psi_{kt}^a - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) - \sum_{k,t,a} h_k \, (\rho_{kt}^a - d_{kt} \psi_{kt}^a) \quad (18)$$

subject to:

$$(2), (3), (4), (5), (6), (7), \text{and } (8)$$

$$\psi_{kt}^a, \; \rho_{kt}^a, \; q_{lkt} \geq 0; \; p_{lkt}, \; y_{ljt} \in \{0, 1\}$$

The only difference to the model presented in Sect. 2 is reflected in the dependency of the revenue to the age of the delivered products. Comparing with objective function (1), objective function (18) has a revenue term that is function of the age of the products sold. Remark that this is a straightforward extension from the base model (Sect. 2), because we have already incorporated a detailed demand fulfillment decision variable (ψ_{kt}^a) tracking the age of the products.

5.2 Demand Parameter

In this model we assume that according to the information about the customer purchasing behaviour, it is possible to determine a parameter $\hat{d}_{kt}^a$ for the demand for

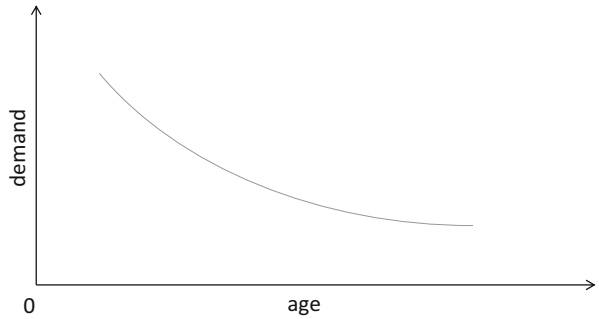


Fig. 4 Schematic example of the age dependent demand

product k with age a in period t. Furthermore, we assume that the demand decreases with the ageing of the products (Fig. 4). For understanding how this parameter may be generated using empirical data about products and customers' willingness to pay the readers are referred to Amorim et al. [3], Tsiros and Heilman [13].

The production planning model of food products with an extended demand parameter (DP-PP-FP) reads:

DP-PP-FP

$$\max \sum_{k,t,a} p_k \, \hat{d}^0_{kt} \, \psi^a_{kt} - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) - \sum_{k,t,a} h_k \, (\rho^a_{kt} - \hat{d}^0_{kt} \psi^a_{kt}) \quad (19)$$

subject to:

$$\hat{d}^0_{kt} \psi^a_{kt} \leq \hat{d}^a_{kt} \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} \quad (20)$$

$$\rho^a_{kt} = \rho^{a-1}_{k,t-1} - \hat{d}^0_{k,t-1} \psi^{a-1}_{k,t-1} \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} \setminus \{0\} \quad (21)$$

(2), (4), (5), (6), (7), and (8)

$$\psi^a_{kt}, \; \rho^a_{kt}, \; q_{lkt} \geq 0; \; p_{lkt}, \; y_{ljt} \in \{0, 1\}$$

The formulation incorporating different demand levels according to the age of the product is very similar to the base model presented in Sect. 2, but in this model the demand parameter is replaced by an extended form that differentiates between products with different ages. Moreover, constraints (20) do not allow the quantity of sold products of a given age to be above the demand curve derived for the respective product (cf. Fig. 4).

6 Illustrative Example

The aim of this illustrative example is to understand the changes in the structure of the production plans when using the different models presented through Sects. 2, 3, 4, and 5.

The setting for the illustrative example consists in a production line ($L = 1$) that has to produce 2 blocks ($N = 2$), each with two products ($K = 4$). For all products/blocks $e_{lk} = 1$, $m_{lj} = 3$ and $\underline{s}_{lj} = \underline{\tau}_{lj} = 1$. For Block 1 the setup cost ($\bar{s}_{lj}$) and the setup time ($\bar{\tau}_{lj}$) is 5 and 1, respectively. For Block 2 $\bar{s}_{lj} = 5$ and $\bar{\tau}_{lj} = 2$. The considered planning horizon has 4 periods ($T = 4$) and the capacity C_{lt} equals 35 for all periods and lines. The remaining parameters are given in Table 1.

We further consider, for the model of Sect. 4.1, that discarding costs $\bar{p}_k$ equal to p_k. In order to obtain one solution for the multi-objective model presented in Sect. 4.2, a weight of 200 was given to the freshness objective in order to have high freshness standards. In general, this is a parameter obtain in pre-computational experiments and it is dependent on the instances. With this weighted linear scalarizing factor, the problem objectives are aggregated in a single one. However, notice that in order to take full advantage of the multi-objective model and obtain the Pareto front a different method, such as the epsilon-constraint approach should be used instead. In the case in which a decreasing value is considered for the price paid for the product throughout its shelf-life (Sect. 5.1), we consider that for products with an age higher than 0, $\hat{p}_k^a = 1$. Finally, for the last model (Sect. 5.2), all products suffer from a 50 % rate of decrease in the demand for each period of ageing ($\hat{d}_{kt}^{a+1} = 0.5\hat{d}_{kt}^{a}$).

6.1 Results and Discussion

Table 2 shows the results for the key decision variables under analysis q_{lkt}, ρ_{kt}^a, ψ_{kt}^a (production, inventory, demand fulfillment) for all models from Sects. 2, 3, 4, and 5. All instances were solved to optimality in less than two seconds by the solver IBM ILOG CPLEX 12.4 and the models were coded in the IBM ILOG OPL IDE. We purposely omitted the objective function values as they are not relevant for our discussion.

Table 1 Remaining parameters for the illustrative example

						d_{kt}			
Block	Product	u_k	p_k	c_{lk}	h_k	1	2	3	4
1	1	1	3	1	0.2	5	0	5	5
1	2	1	3	1	0.2	0	10	10	5
2	3	2	4	2	0.1	10	5	0	10
2	4	3	4	2	0.1	5	15	10	5

Table 2 Results for the decision variables q_{lkt}, ρ^a_{kt}, ψ^a_{kt} for all models (Sects. 2, 3, 4, and 5). Values that are the same across all models are in light grey and results that differ between the models (except the base one – B-PP-FP) are in bold. Notice that the relation between the time and age index is respected in this table ($a \in \mathscr{A} = \{a \in \mathbb{Z}_0^+ | a \leq t-1\}$)

		q_{lkt}				ρ^a_{kt}										ψ^a_{kt}								
	Period *(t)*	1	2	3	4	1	2	2	3	3	3	4	4	4	4	1	2	2	3	3	3	4	4	4
Model	Product *(k)* \Age *(a)*	0	0	0	0	0	0	1	0	1	2	0	1	2	3	0	0	1	0	1	2	0	1	2
B-PP-FP (Base)	1	5	0	10	0	5	0	0	10	0	0	0	5	0	0	100%	0%	0%	100%	0%	0%	0%	100%	0%
	2	8	0	15	0	8	0	8	15	0	0	0	5	0	0	0%	0%	80%	100%	0%	0%	0%	100%	0%
	3	10	5	0	10	10	5	0	0	0	0	10	0	0	0	100%	100%	0%	0%	0%	0%	100%	0%	0%
	4	5	25	0	5	5	25	0	0	10	0	5	0	0	0	100%	100%	0%	0%	100%	0%	100%	0%	0%
IE-PP-FP (Inventory expiry)	1	5	0	5	5	5	0	0	5	0	0	5	0	0	0	100%	0%	0%	100%	0%	0%	100%	0%	0%
	2	0	10	10	5	0	10	0	10	0	0	5	0	0	0	0%	100%	0%	100%	0%	0%	100%	0%	0%
	3	**15**	**0**	0	10	**15**	**0**	**5**	0	0	0	10	0	0	0	100%	**0%**	**100%**	0%	0%	0%	100%	0%	0%
	4	**9**	**20**	**0**	5	**9**	**20**	**4**	**0**	**9**	**0**	5	0	0	0	100%	**73%**	**27%**	**0%**	**90%**	**0%**	100%	0%	0%
CB-PP-FP (Consumer behaviour)	1	5	0	5	5	5	0	0	5	0	0	5	0	0	0	100%	0%	0%	100%	0%	0%	100%	0%	0%
	2	0	10	10	5	0	10	0	10	0	0	5	0	0	0	0%	100%	0%	100%	0%	0%	100%	0%	0%
	3	**15**	**0**	0	10	**15**	**0**	**5**	0	0	0	10	0	0	0	100%	**0%**	**100%**	0%	0%	0%	100%	0%	0%
	4	**9**	**20**	**0**	5	**9**	**20**	**4**	**0**	**5**	**4**	5	0	0	0	100%	**100%**	**0%**	**0%**	**50%**	**40%**	100%	0%	0%
DC-PP-FP (Discarding costs)	1	5	0	5	5	5	0	0	5	0	0	5	0	0	0	100%	0%	0%	100%	0%	0%	100%	0%	0%
	2	0	10	10	5	0	10	0	10	0	0	5	0	0	0	0%	100%	0%	100%	0%	0%	100%	0%	0%
	3	**15**	**0**	0	10	**15**	**0**	**5**	0	0	0	10	0	0	0	100%	**0%**	**100%**	0%	0%	0%	100%	0%	0%
	4	**9**	**20**	**0**	5	**9**	**20**	**4**	**0**	**9**	**0**	5	0	0	0	100%	**73%**	**27%**	**0%**	**90%**	**0%**	100%	0%	0%
MF-PP-FP (Measuring freshness)	1	5	0	5	5	5	0	0	5	0	0	5	0	0	0	100%	0%	0%	100%	0%	0%	100%	0%	0%
	2	0	10	10	5	0	10	0	10	0	0	5	0	0	0	0%	100%	0%	100%	0%	0%	100%	0%	0%
	3	**10**	**5**	0	10	**10**	**5**	**0**	0	0	0	10	0	0	0	100%	**100%**	**0%**	0%	0%	0%	100%	0%	0%
	4	**6**	**14**	**10**	5	**6**	**14**	**1**	**10**	**0**	**0**	5	0	0	0	100%	**93%**	**7%**	**100%**	**0%**	**0%**	100%	0%	0%
VF-PP-FP (Value of freshness)	1	5	0	5	5	5	0	0	5	0	0	5	0	0	0	100%	0%	0%	100%	0%	0%	100%	0%	0%
	2	0	10	10	5	0	10	0	10	0	0	5	0	0	0	0%	100%	0%	100%	0%	0%	100%	0%	0%
	3	**10**	**5**	0	10	**10**	5	**0**	0	0	0	10	0	0	0	100%	**100%**	**0%**	**0%**	0%	0%	100%	0%	0%
	4	**5**	**14**	**10**	5	**5**	14	**0**	**10**	**0**	**0**	5	0	0	0	100%	**93%**	**0%**	**100%**	**0%**	**0%**	100%	0%	0%
DP-PP-FP (Demand parameter)	1	5	0	5	5	5	0	0	5	0	0	5	0	0	0	100%	0%	0%	100%	0%	0%	100%	0%	0%
	2	0	10	10	5	0	10	0	10	0	0	5	0	0	0	0%	100%	0%	100%	0%	0%	100%	0%	0%
	3	**10**	**5**	0	10	**10**	**5**	**0**	0	0	0	10	0	0	0	100%	**100%**	**0%**	0%	0%	0%	100%	0%	0%
	4	**6**	**14**	**10**	5	**6**	**14**	**1**	**10**	**0**	**0**	5	0	0	0	100%	**93%**	**7%**	**100%**	**0%**	**0%**	100%	0%	0%

Overall, results seem to indicate that even for an illustrative example, by incrementally introducing different features and methods for better tackling the food production planning, different solutions are obtained for almost every model tested. The only production plan leading to spoiled products is the base model (B-PP-FP) when products 1 and 2 reach an age of 1 in period 4 ($\rho^1_{14} = \rho^1_{24} = 5$). All the other models are able to avoid the expiration of these products due to different reasons. For example, while the model considering inventory expiry (IE-PP-FP) avoids spoilage by the fact that we limit the demand fulfilment to products with a significant remaining shelf-life, the model introducing discarding costs (DC-PP-FP) is able to achieve the same solution by penalizing the occurrence of expired inventory.

One interesting analysis lies on the different solutions found with the inventory expiry model (IE-PP-FP) and the consumer behaviour one (CB-PP-FP). The only difference between these models in the inclusions of constraints (12) and (13) in the CB-PP-FP model, which mimic the fact that customers pick up the fresher available products. For product 4 in period 2 when 20 units are produced in both models ($q_{142} = 20$), model IE-PP-FP is able to allocate in order to satisfy demand part of the production of period 2 and part of the production of period 1 with age 1 ($\psi^0_{42} = 73\,\%$ and $\psi^1_{42} = 27\,\%$). On the contrary, the CB-PP-FP model is forced to satisfy all demand in period 2 with the production executed in the same day ($\psi^0_{42} = 100\,\%$ and $\psi^1_{42} = 0\,\%$). These differences ultimately lead to the fact that customers in period 3 are penalized in the CB-PP-FP model as they will be satisfied with less fresh products ($\psi^2_{43} = 40\,\%$). This fact could potentially lead to lost sales and it reflects the importance of proper inventory control when dealing with perishable products.

From the seven models, it is clear that the last three are able to better incorporate the consumer eagerness for fresher products. In particular, the model measuring freshness (MF-PP-FP) and the model having an extended demand parameter (DP-PP-FP) have an equivalent behaviour. Both models incorporate explicitly the importance of satisfying customers with a high degree of freshness. The difference between them relies on the amount and quality of information the decision maker has when setting up the model (less information for the MF-PP-FP and more for the DP-PP-FP).

7 Risk-Conscious Planning

In the previous models (Sects. 2, 3, 4, and 5) a major assumption is the deterministic parameter of demand. As seen in the illustrative example (Sect. 6), in this setting spoiled products will only appear in case no perishability considerations are taken into account. This can be done by constraining the domain of the variables used to track both demand fulfillment and inventory levels. However, in this type of industries, producers and retailers struggle with significant amounts of spoiled products. These quantities are tightly correlated to the uncertainty in the forecast of

demand. Explicitly acknowledging the existence of such uncertainty and adopting a risk conscious planning promise robust and sustainable gains. With this approach the distribution of the gains is sharper and further away from the loss side. This comes at the expense of a decrease in the expected profit.

In this section we start by extending the Base model (Sect. 2) in order to cope with an uncertain demand parameter and we then give an example of a risk-averse formulation that tackles explicitly the conditional value-at-risk. The section ends with an extension of the illustrative example of Sect. 6.

7.1 Risk-Neutral Model

The uncertainty of the demand parameter $\tilde{d}_{kt}^{v}$ may be modeled though a set of scenarios $\mathscr{V}$ that have a probability of occurrence ϕ_v. In order to incorporate this stochastic parameter into the formulation, it is necessary to determine the moment in time in which demand is unveiled with certainty. In the most common setting, the planner has to decide about the sizing and scheduling of lots in the first-stage and then inventory allocation decisions are done with full knowledge of the demand parameter (second-stage).

To model the production planning of perishable foods good in an uncertain setting it is necessary to define the following second-stage decision variables:

Second-Stage Decision Variables

$\tilde{\rho}_{kt}^{av} \geq 0$ initial inventory of product k with age a available at period t in scenario v

$\tilde{\psi}_{kt}^{av} \geq 0$ fraction of the maximum demand for product k delivered with age a at period t in scenario v

The risk-neutral production planning model of food products (RN-PP-FP) reads:

RN-PP-FP

$$\max \sum_{v} \phi_v [\sum_{k,t,a} p_k \, \tilde{d}_{kt}^{v} \, \tilde{\psi}_{kt}^{av} - \sum_{k,t,a} h_k \, (\tilde{\rho}_{kt}^{av} - \tilde{d}_{kt}^{v} \, \tilde{\psi}_{kt}^{av})] - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt} - \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) \tag{22}$$

subject to:

$$\sum_{a} \tilde{\psi}_{kt}^{av} \leq 1 \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, v \in \mathscr{V} \tag{23}$$

$$\tilde{\rho}_{kt}^{av} = \tilde{\rho}_{k,t-1}^{a-1,v} - \tilde{d}_{k,t-1}\tilde{\psi}_{k,t-1}^{a-1,v} \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, a \in \mathscr{A} \setminus \{0\}, v \in \mathscr{V} \tag{24}$$

$$\sum_{l} q_{lkt} = \tilde{\rho}_{kt}^{0v} \quad \forall k \in \mathscr{K}, t \in \mathscr{T}, v \in \mathscr{V} \tag{25}$$

(5), (6), (7), and (8)

$$\tilde{\psi}_{kt}^{av},\ \tilde{\rho}_{kt}^{a},\ q_{lkt} \geq 0;\ p_{lkt},\ y_{ljt} \in \{0, 1\} \tag{26}$$

The objective function (22) maximizes the expected profit of the producer over the planning horizon. In this two-stage stochastic formulation, both revenue and holding costs are now dependent on the scenario realization. The second-stage constraints related to inventory management and demand fulfillment (23), (24), and (25) were changed to incorporate the new stochastic setting.

7.2 Risk-Averse Model

In face of uncertainty the planner may take several attitudes in terms of risk. For a risk-conscious attitude it is necessary to introduce a risk measure into the formulation. Recent studies showed that for production planning of perishable food goods the conditional value-at-risk [10, 11], which is very used in portfolio optimization, is a good option as it reduces drastically the amount of expired products at the expense of a small loss on the expected profit [1]. To introduce this risk measure in the formulation we need to further define two decision variables and two parameters α and λ. α controls the confidence interval of the conditional value-at-risk and λ controls the risk-aversion emphasis of the generated plan.

Conditional Value-at-Risk Decision Variables

η value-at-risk
δ_v auxiliary variable for calculating the conditional value-at-risk

In Fig. 5 a graphical interpretation of this measure is given. Consider X to be a random profit distribution, from the figure it is easy to interpret that the conditional value-at-risk (cVaR) is then defined as $\mathbb{E}[X|X \leq VaR(X)]$.

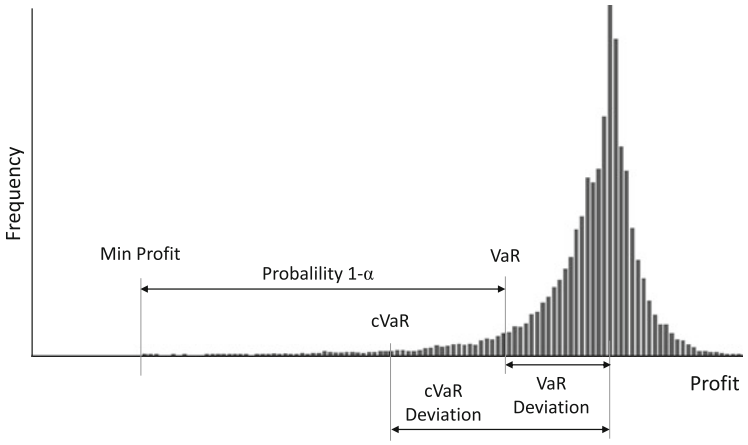


Fig. 5 Graphical interpretation of the conditional value-at-risk measure (Adapted from Sarykalin et al. [12])

The risk-conscious production planning model of food products (RC-PP-FP) reads:

RC-PP-FP

$$\max \sum_{v} \phi_v [\sum_{k,t,a} p_k \, \tilde{d}^v_{kt} \, \tilde{\psi}^{av}_{kt} - \sum_{k,t,a} h_k \, (\tilde{\rho}^{av}_{kt} - \tilde{d}^v_{kt} \, \tilde{\psi}^{av}_{kt})] - \sum_{l,j,t} \bar{s}_{lj} \, y_{ljt}$$

$$- \sum_{l,k,t} (\underline{s}_{lk} \, p_{lkt} + c_{lk} \, q_{lkt}) + \lambda (\eta - \frac{1}{1-\alpha} \sum_{v} \phi_v \, \delta_v) \tag{27}$$

subject to:

$$\text{(23), (24), (25), (5), (6), (7), and (8)}$$

$$\delta_v \geq \eta - (\sum_{k,t,a} p_k \, \tilde{d}^v_{kt} \, \tilde{\psi}^{av}_{kt} - \sum_{k,t,a} h_k \, (\tilde{\rho}^{av}_{kt} - \tilde{d}^v_{kt} \, \tilde{\psi}^{av}_{kt})) \quad v \in \mathcal{V} \tag{28}$$

$$\tilde{\psi}^{av}_{kt}, \; \tilde{\rho}^{a}_{kt}, \; q_{lkt}, \; \delta_v \geq 0; \; \eta \in \mathbb{R}; \; p_{lkt}, \; y_{ljt} \in \{0, 1\} \tag{29}$$

The objective function (27) maximizes the expected profit and, simultaneously, it maximizes the conditional value-at-risk with a confidence of α. The second-stage constraints (23), (24), and (25) are the same of the risk-neutral model. A new constraint (28) has to be added to attribute the variable δ_v a value of zero, if scenario v yields a profit higher than η. Otherwise, variable δ_v is given the difference between the value-at-risk η and the corresponding second-stage profit $\sum_{k,t,a} p_k \, \tilde{d}^v_{kt} \, \tilde{\psi}^{av}_{kt} - \sum_{k,t,a} h_k \, (\tilde{\rho}^{av}_{kt} - \tilde{d}^v_{kt} \, \tilde{\psi}^{av}_{kt})$.

Table 3 Profit values for the Risk-neutral model, and a Risk-averse model

		Scenario Profit			
	Expected Profit	Low	Medium	High	
RN-PP-FP	105.8	0.0	157.0	161.2	
RC-PP-FP λ=4	95.3	37.8	122.8	126.0	

7.3 Extended Illustrative Example

To understand the importance of a risk-conscious production planning of perishable food products let us use the illustrative example presented in Sect. 6. We have extended the data set by distinguishing three possible demand scenarios, all with the same probability of occurrence (0.33). A medium scenario in which the demand is equal to the one presented in Table 1, a low one in which demand is 50 % of its expected value and, finally, a high one in which demand is 50 % higher than in the medium scenario. Therefore, we use a simplified scenario-tree with only three scenarios. For assessing the impact of uncertainty and understanding the implications of a risk-conscious planning we obtained optimal solution values for three models: (1) the Base model presented in Sect. 2, (2) the Risk-neutral model RN-PP-FP presented in Sect. 7.1, and (3) a Risk-averse model based on RC-PP-FP presented in Sect. 7.2 (setting $\lambda = 4$ and $\alpha = 0.95$). Results for the last two models are presented in Table 3. The Base model (B-PP-FP) has a solution with a profit of 157.0.

Results indicate that with the stochastic demand parameter the expected profit drops considerably. Notice that demand parameter used in the Base model is the expected value of the uncertain demand parameter ($\tilde{d}_{kt}^{v}$). In the Risk-neutral approach there is one scenario that would result in a profit of 0. This "bad" scenario is mitigated by in the Risk-conscious model that has its worst scenario with a profit of 37.8. This more balanced overall solution with less dispersion of the profit distribution comes at the expense of a slightly lower expected value of profit.

8 Conclusions

In this paper, we have reviewed several ways of integrating different challenges related to exogenous factors (such as customer behaviour and the perishable nature of the products) arising in the production planning of food products. The formulations have the same base model as starting point and we have organised them based on the extensions of the model components required: constraints, objective function and parameters. In particular, we have analysed how to limit the inventory age based on an adapted simple plant location reformulation, how to incorporate the

consumer behaviour within the inventory policy, how to include discarding costs in the objective function, how to model customer willingness for fresh products in a multi-objective framework and how to value freshness either in the price or demand parameters. To analyse the implications of each of these "ingredients", an illustrative example is presented and solved, exposing the different solution structures achieved. The differences across the solutions show the importance of choosing an approach suitable to the particular business setting, in order to accommodate the multiple challenges present in these industries. Moreover, acknowledging the perishable nature of the products and evaluating the amount and quality of information at hands may be crucial in lowering disposal costs and achieving higher service levels. There are other ingredients not so related to the perishable nature of food products that are also important in food production planning. For example, Wang et al. [14] deals with the incorporation of batch traceability that is increasingly important with the recent cases of products recall.

In the last Section, we analyzed a recent trend in supply chain planning – risk-conscious planning. The mitigation of uncertainties in this industries is crucial since their effects are leveraged by the perishable nature of the products. The importance of a risk-averse approach is especially noticeable in terms of avoiding disastrous uncertain outcomes.

Future work should explore these extensions from a computational point of view. Therefore, devising which solution methods are more appropriate for each setting is still a gap to be addressed.

Acknowledgements This work is financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "FCOMP-01-0124-FEDER-041499" and project "NORTE-07-0124-FEDER-000057".

References

1. Amorim, P., Alem, D., Almada-Lobo, B.: Risk management in production planning of perishable goods. Ind. Eng. Chem. Res. **52**, 17538–17553 (2013)
2. Amorim, P., Antunes, C.H., Almada-Lobo, B.: Multi-objective lot-sizing and scheduling dealing with perishability issues. Ind. Eng. Chem. Res. **50**, 3371–3381 (2011)
3. Amorim, P., Costa, A., Almada-Lobo, B.: Influence of consumer purchasing behaviour on the production planning of perishable food. OR Spectr. **36**(3), 669–692 (2014)
4. Amorim, P., Meyr, H., Almeder, C., Almada-Lobo, B.: Managing perishability in production-distribution planning: a discussion and review. Flex. Serv. Manuf. J. **25**, 389–413 (2013)
5. Amorim, P., Pinto-Varela, T., Almada-Lobo, B., Barbósa-Póvoa, A.: Comparing models for lot-sizing and scheduling of single-stage continuous processes: operations research and process systems engineering approaches. Comput. Chem. Eng. **52**, 177–192 (2013)
6. Bilde, O., Krarup, J.: Sharp lower bounds and efficient algorithms for the simple plant location problem. Ann. Discret. Math. **1**, 79–97 (1977)

7. Fleischmann, B., Meyr, H., Wagner, M.: Advanced planning. In: Stadtler, H., Kilger, C. (eds.) Planning Supply Chain Management and Advanced Planning, 4th edn., pp. 81–106. Springer, Berlin/Heidelberg (2008)
8. Günther, H.-O., Grunow, M., Neuhaus, U.: Realizing block planning concepts in make-and-pack production using MILP modelling and SAP APO. Int. J. Prod. Res. **44**, 3711–3726 (2006)
9. Pahl, J., Voß, S.: Integrating deterioration and lifetime constraints in production and supply chain planning: a survey. Eur. J. Oper. Res. **238**(3), 654–674 (2014)
10. Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. J. Risk **2**, 21–42 (2000)
11. Rockafellar, R.T., Uryasev, S.: Conditional value-at-risk for general loss distributions. J. Bank. Financ. **26**, 1443–1471 (2002)
12. Sarykalin, S., Serraino, G., Uryasev, S.: Value-at-risk vs. conditional value-at-risk in risk management and optimization. In: Tutorials in Operations Research INFORMS, Hanover, pp. 270–294 (2008)
13. Tsiros, M., Heilman, C.M.: The effect of expiration dates and perceived risk on purchasing behavior in grocery store perishable categories. J. Mark. **69**, 114–129 (2005)
14. Wang, X., Li, D., O'Brien, C.: Optimisation of traceability and operations planning: an integrated model for perishable food production. Int. J. Prod. Res. **47**, 2865–2886 (2009)

Sectors and Routes in Solid Waste Collection

Ana M. Rodrigues and J. Soeiro Ferreira

Abstract Collecting and transporting solid waste is a constant problem for municipalities and populations in general. Waste management should take into account the preservation of the environment and the reduction of costs. The goal with this paper is to address a real-life solid waste problem. The case reveals some general and specific characteristics which are not rare, but are not widely addressed in the literature. Furthermore, new methods and models to deal with sectorization and routing are introduced, which can be extended to other applications. Sectorization and routing are tackled following a two-phase approach. In the first phase, a new method is described for sectorization based on electromagnetism and Coulomb's Law. The second phase addresses the routing problems in each sector. The paper addresses not only territorial division, but also the frequency with which waste is collected, which is a critical issue in these types of applications. Special characteristics related to the number and type of deposition points were also a motivation for this work. A new model for a Mixed Capacitated Arc Routing Problem with Limited Multi-Landfills is proposed and tested in real instances. The computational results achieved confirm the effectiveness of the entire approach.

1 Introduction

Collecting and transporting solid waste is a common problem for municipalities and populations in general. A good waste management program takes into account routing issues, not only because the economic benefits, but also to preserve the

A.M. Rodrigues (✉)
ISCAP-Instituto Politécnico do Porto, Rua Jaime Lopes Amorim, 4465-004 S. Mamede de Infesta, Portugal

INESC TEC, Porto, Portugal
e-mail: amr@inesctec.pt

J.S. Ferreira
Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias 4200-465 Porto, Portugal

INESC TEC, Porto, Portugal
e-mail: jsf@inesctec.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_20

environment. Better routes mean lower fuel consumption, which leads to lower emissions of CO_2. Collecting vehicles have an extremely high level of consumption, and reducing a few km on a daily route represents a significant reduction of costs by the end of the month.

The aim of this paper is to address a real-life situation, based on the region of Monção in the North of Portugal. The case study reveals some general and specific characteristics which are not rare, at least in the country.

Municipalities are so large that prior sectorization is convenient, which means dividing the territory into sectors. Electromagnetism and Coulomb's Law were used to divide the territory. Forces of attraction or repulsion were used to group the elementary regions (in Portugal these regions are called *freguesias*) into sectors. Besides considering the common facets of sectorization, this new method also addresses particular situations and requirements of *freguesias*, as in some of them waste is collected daily, while in others it is collected two or three times a week. The case study also features specific conditions related to the number and kind of deposition points. Two types of points are considered: landfills and transfer stations. Landfills can be used whenever a collecting vehicle needs to be emptied and they have no limitation regarding the number of visits. Transfer stations are points that temporarily receive waste. What happens is that sometimes transfer stations are small and cannot receive all the waste. As a consequence, the number of daily visits can be limited. To deal with these matters, a new problem and model are introduced: The Mixed Capacitated Arc Routing Problem with Limited Multi-Landfills (MCARP-*LML*). Sectorization and Routing are used to solve the case of solid waste collection.

The paper is organized as follows: Sect. 2 briefly reviews the literature on sectorization and applications. Section 3 provides an overview of routing, particularly arc routing problems with capacity restrictions. The real-case of waste collection in Monção is described in Sect. 4. Section 5 presents the solution approach proposed to solve the problem. Section 6 provides the computational results based on real instances from Monção. Finally, Sect. 7 provides some conclusions.

2 Sectorization: Reasons and Scope

Sectorization means dividing into sectors or parts, a procedure that occurs in many contexts and applications, usually to achieve some goal or to facilitate an activity. Most of the time, this division or partition aims at better organizing or simplifying a large problem into smaller sub-problems, or promoting groups with similar characteristics. The idea is different from the one of a clustering process in which, although the groups to be formed are composed of individuals with similar features, they must be as different as possible from each other, [23].

The suitability of sectorization is not new. One of the initial publications, [21], discusses the division of a territory applied to the definition of political districts.

More recently, the literature refers to various other applications, such as: dividing sales territories by different sellers, school districts, salt spreading operations in the winter, or collecting waste. Some applications are described below.

In order to evaluate the quality of the sectors (districts, parts) obtained after sectorization, it is convenient to use some general measures:

- Equilibrium – different districts must contain approximately the same "population" or "amount of work";
- Contiguity – each district or sector must be composed of just "one body", that is, it should be possible to "move" between any pair of points in a district without leaving the district;
- Compactness – it is a measure of "concentration", that is, "U" shapes and dispersed territory should be avoided; instead, round shapes that are more compact should be preferred.

Sectorization related to political districting has been studied for some decades. The main idea is dividing a certain territory into a given number of sectors (districts), based on the principle of "one man – one vote". Votes are transformed into seats, and a correct partition of the territory cannot admit advantages and/or disadvantages for some sectors. Other measures are considered in political districting: the integrity of territories, the respect for the administrative subdivision of the territory, the existence of small communities, and the preservation of the minorities' strength. In [43], the authors organized the literature related to political districting in groups according to the models and approaches used. Other references are [14, 21] or [6].

Sectorization is also used to design sales territories. Changes in the number of salesmen can not only justify a redefinition of territories, but also a better use of the existing potential or a better coverage of the territory. The main idea is defining boundaries within a territory to produce sectors. The common objective is to maximize the profit, dividing a certain "sales force" into a given number of smaller areas. The work [47] analyzes some approaches to maximize the profit described by different authors in the literature. Other references are: [20, 23] or [51].

School districting or redistricting consists of dividing a city (or a municipality) into smaller areas or neighborhoods that must be assigned to schools. A good sectorization makes it possible, for example, to minimize the total travel distance between student residences and schools, and/or to take into account racial balance or crossing arterial roads on the way home from school. The papers [49] and [7] present interesting results on school districting.

Sectorization can also be used to partition a certain region into smaller areas to ensure an optimal allocation of health services. Optimal hospital districting, considering demand and capacity measured in number of hospital beds, is mentioned in [41]. The work in [4] presents a problem linked to home health care. Another reference, [37], deals with the partitioning of a territory into a number of areas where at least one source of a certain social service must be present.

Sectorization (districting or redistricting) is also present in many other situations: the definition of police command boundaries is tackled in [9]; medical emergency

systems in highways are addressed in [22]; cost minimization in jail systems, taking into account existing and possible new jails, are described in [36]; the authors of [5] deal with urban emergency services such as fire services; sector design related to patrolling operations by vessels of a maritime agency is described in [34]; in [15], the authors determine an overall service area and the necessary transit mobile repair units in emergency situations.

2.1 Sectors and Waste Collection

Sectorization has also been applied to Solid Waste Collection. In [19], the authors propose to create balanced sectors according to the daily collection time. Firstly, the region is divided into a predetermined number of sectors. Then, and for each sector, collector vehicle routes are built so that the time spent in each sector is limited in the pre-defined range $[T_{min}, T_{max}]$, and as closely as possible to a "target" $T^* \in [T_{min}, T_{max}]$. The number of connected components in each sector is also reduced. The work [35] reports a situation in which streets without direction restrictions are grouped into sectors. The amount of waste to be collected in each sector should not exceed the vehicle's capacity. The objective is to minimize the total cost associated with revisited streets or streets where collection is not required. Work levels are used in [29] to solve sectoring problems in urban waste collection. Two vertices belong to level n if the shortest route between them has n intermediate nodes. A feasible solution for sectoring rejects solutions with very high work levels. The authors in [39] introduce three methods to deal with municipal waste collection: two different two-phase methods, where phase 1 is dedicated to creating the sectors, and phase 2 consists of determining the trips in each sector, and a third method based on best insertion. Evaluation criteria such as imbalance, diameter and dispersion measures are used to compare the algorithms.

3 Routing: Overview and Applications

General routing problems are addressed differently in the literature. Using graph language, there are not only routes associated with arcs/edges, but also routes associated with vertices, and more general routes associated with both arcs/edges and vertices. Arc Routing Problems (ARP) are related to the determination of a least cost traversal of the set (or subset) of edges, on a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. This graph can be undirected, directed (using the term arc instead of edge), or mixed (with edges and arcs). A cost $c_{ij} \geq 0$ is associated with each edge $(i, j), i \neq j$. If $(i, j) \notin E$, then $c_{ij} = \infty$. These problems appear in a large variety of practical contexts ([10, 11, 38] and [45]), such as mail delivery, delivery of telephone books, garbage collection, street sweepers, salt gritting, inspection of streets for maintenance, meter reading, snow

removal, internet routing, cutting process manufacturing, or printed circuit board manufacturing. Node Routing Problems (NRP) are analogous problems but they focus on vertices (or nodes) instead of edges. An NRP can be transformed into an ARP and vice versa, [2]. Well-known examples are the Travelling Salesman Problem (TSP), the Asymmetric TSP and the Vehicle Routing Problem (VRP). ARP and NRP are special cases of a broader class of problems named General Routing Problems (GRP). The GRP, introduced by [40], look for the minimum cost tour on a graph $G = (V, E)$, where a subset of edges E_R ($E_R \subseteq E$) and a subset of vertices V_R ($V_R \subseteq V$) are mandatory.

Despite the importance of the NRP, more broadly referred in the literature, this paper only addresses arc routing.

According to the characteristics of the problems, ARP can be generally classified into four independent/distinct major classes: Windy, Capacitated, Hierarchical and Other problems. The latter includes all the problems that do not fit the other three classes. Capacitated ARP (CARP) emerges when capacity restrictions are introduced into the closed paths of each vehicle (or postman).

CARP can be defined in a graph $G = (V, E)$, undirected, directed or mixed, with a subset E_R ($E_R \subseteq E$) of required edges and a special vertex, v_0, called depot. A demand $d_e > 0$ is associated with each required edge $e \in E_R$. A fleet of n vehicles (not necessarily identical), each one with capacity $Q_i > 0, i = 1, \ldots, n$, is available. The objective is to find a minimum cost set of vehicle routes, which must start and end at the depot, such that each required edge is serviced exactly once, and the sum of demands of the serviced edges, in each route, does not exceed the vehicle's capacity, [30] and [33]. The CARP is an NP-Hard problem and it was introduced by [17]. A recent survey of CARP and its variants is provided by [50].

The CARP is also a generalization of the Capacitated Chinese Postman Problem (CCPP), where demand is positive in all edges, [12].

The Mixed CARP (MCARP) is a CARP with arcs and edges, that is, with directed and undirected links, respectively. The Periodic CARP (PCARP), defined by [25], is motivated by the need to assign a daily service to the edges, in real applications. It is an extension of the CARP to multiple periods, [8]. The final objective is to minimize the required fleet and the total cost of the trips in the selected multi-period horizon.

The Location ARP (LARP), initially called Arc Oriented Location Routing by [31], is another extension of ARP. LARP simultaneously deals with location and arc routing decisions. The work in [42] addresses applications in distribution systems, while [32] contains a survey and suggestions for future research.

CARP with Refill Points (CARP-RP) are LARP with two different types of vehicles: the servicing vehicle and the refilling vehicle. The refilling vehicle can meet the first one at any place to refill it. Amaya et al. [1] presents a practical application of this problem: "road network maintenance, where the road markings have to be painted or repainted every year." In the Sectoring ARP (SARP), the network is partitioned into a given number K of sectors. The aim is to solve K MCARP and to minimize the total duration of the trips over all sectors, [39]. Special applications are associated with waste collection in large urban areas.

The Stochastic CARP (SARP) is a stochastic version of the CARP with random demand on the arcs, [13]. Real applications may be found in waste collection.

CARP with time dependent service costs are a variant of CARP in which the cost of servicing some arcs depends on the time the service starts, [48]. The applications include spreading grit on icy roads, and the timing of interventions is crucial.

Another CARP is the Extended CARP (ECARP) [27], a problem that includes extensions, such as mixed multigraph with edges and arcs and parallel links, deadheading and collection costs per link, prohibited turns and turn penalties, and an upper limit on the cost of any trip.

In the CARP with Intermediate Facilities (CARPIF), vehicles may unload or replenish at intermediate facilities, which are a subset of the vertices. Garbage collection with visits to dump sites or incinerators is a possible application. An equivalent situation is the one in which the vehicle makes deliveries instead of collections, and vehicles must replenish to meet the demand, [16].

The CARP with Unit Demand (CARPUD) is a particular CARP where all required edges present unit demand, [3].

A Multi-objective version of the CARP is presented in [28], and its goalis two-fold: minimizing the total duration of the trips, and reducing the duration of the longest trip. The work presented in [26] gives the example of waste management companies interested in both balancing the trips and minimizing their total durations.

The VRP is a capacitated version of the TSP, as the CARP is the capacitated version of the Rural Postman Problem, [2]. One approach to the CARP is the transformation into a VRP.

4 Real Case Study

Collecting and transporting solid waste is a difficult and complicated problem in modern societies. In this task, there are often many different specificities and constraints that need to be taken into account.

The case study presented in this paper is based on a municipality in the North of Portugal, Monção, which has some specific characteristics that are not rare, at least in the country.

This is an outcome of the work that the authors did on waste collection problems in Monção. The municipality of Monção is a region with 220 km^2 and a population of about 20,000 inhabitants. This municipality is a combination of rural and urban areas, and the population is divided into 33 small regions, called *freguesias* in Portuguese (see the map in Fig. 1). This is a region with a strong component of emigrants and in the summer the population increases. As a consequence, the amount of waste produced increases as well. In the “more rural” regions, waste must be collected 2 or 3 times a week (depending on the season), while in “more urban”, regions waste must be collected every day.

There are two deposition points in Monção: the landfill of Valença and the transfer station of Messegães. The transfer station of Messegães also receives waste from other municipalities. Inhabitants also use this transfer station to deposit large

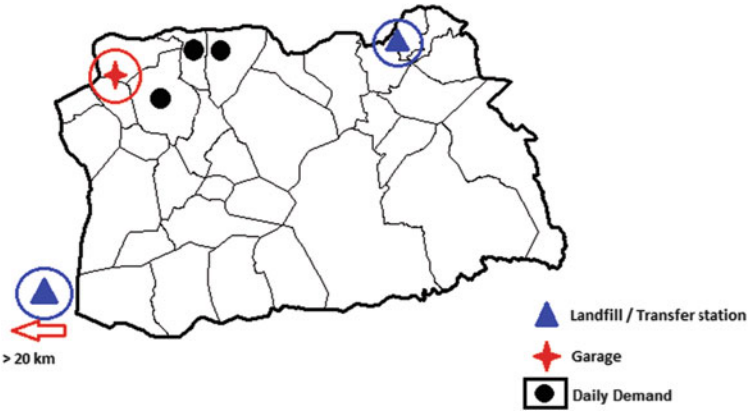


Fig. 1 All the *freguesias* of Monção, the garage, waste disposal points and *freguesias* with daily demand

objects such as old furniture. Generally, this transfer station only allows a single visit a day per vehicle. The municipality has around 1,600 containers of different types, from simple trash bags to more modern and large containers. An estimation was made of the time a container is collected, which depends on its type. The average speed of the vehicle was also estimated (30 km/h). The volume of each container was calculated considering the fact that, on average, containers are not completely full and vehicles have a system that compresses the waste collected. This paper only considers the case of one vehicle but in [46] a model with two types of vehicles is presented. Vehicles with different capacities, different costs and other specific characteristics such as dedicated containers.

To summarize, the challenge posed involves:

- a vehicle with a limited capacity;
- a heterogeneous group of containers (different capacities, different time to collect);
- one-way streets;
- different collecting frequencies (every day or 2 or 3 times a week);
- the garage is not a deposition point (when the vehicles start and end at the garage, vehicles are empty);
- some deposition points can present limitations in terms of the number of visits.

5 Two-Phase Approach

A two phase approach has been developed to solve the Monção case study. The process starts with a sectorization phase where the territory of that municipality is divided into smaller sectors to be considered in the next phase (circuits).

Sectorization deals with points: each elementary unit represents the amount of waste produced by a community. The second phase of the approach deals with route planning. A route will emerge for each sector obtained during the first phase. The routing process is focused on the requirement of serving the link (edge or arc). There are several collecting points along the link and, in some cases, the collection is door to door. Routes must meet some specific constraints, such as the transfer station's limitation in terms of the amount of waste received. If the resulting routes are not "good enough" new sectors will be produced, and, after that, new routes will be obtained.

5.1 Sectors: Solution Based on Electromagnetism

There are two essential reasons for building sectors right from the start: firstly, the municipality of Monção is so large that it is not possible to collect all the waste in just one circuit, and for that reason waste must be collect throughout the week. The second reason is related to the frequency of collection. Different *freguesias* present different demands. Throughout most of the year, there are *freguesias* where the waste must be collected twice a week, and others where the collection occurs daily. The approach proposed considers the frequency of collection as an input to the sectorization phase.

The sectorization process presented in this paper was inspired by electromagnetism. When regarding the containers over a map, those belonging to the same sector should demonstrate some kind of "attraction"; containers in different sectors should present some kind of "repulsion". Quite simply, this is the idea behind the approach and its connection to electromagnetism and Coulomb's Law, which establishes a relation of force between two point charges, [44].

Adaptations and extensions of the concepts of attraction and repulsion will be conducted to be applied in sectorization.

Coulomb's Law states that:

> The force between a given pair of charges is inversely proportional to the square of the distance between them. [. . .] is directly proportional to the *quantity* of one charge multiplied by the *quantity* of the other, (in [24]).

Suppose there are two electrical charged points with charges q_1 and q_2, which are at a distance of d_{12}. The force $\overrightarrow{F}$ between the two charges presents an intensity given by:

$$\overrightarrow{F} = k \cdot \frac{|q_1| \cdot |q_2|}{d_{12}^2} \cdot \hat{r}_{12}. \tag{1}$$

k represents the Constant of Coulomb and corresponds to $8.99 \times 10^9\,\mathrm{N\,m^2.C^{-2}}$ and $\hat{r}_{12}$ is unit vector.

The force is along the straight line joining the charges. If they have the same sign, the electrostatic force between them is repulsive; if they have different signs, the force between them is attractive.

5.1.1 Attraction Produces Sectors

Inspired by electromagnetism, a system composed of n collecting points is seen as n "charged points" over a map. The position of each point (latitude and longitude) is known and the respective "charge" corresponds to the amount of waste to collect at that point.

Following Coulomb's Law, represented in (1), a symmetric matrix A of attractions between each pair of n collection points was calculated. Initially, each pair of points was always supposed to have charges with different signals, that is, the relation between two points is always attractive, and never repulsive. Furthermore, the constant k was set to be equal to 1.

The vector length that represents the "force of attraction" between two points $I = (x_i, y_i)$ and $J = (x_j, y_j)$ with charges q_i and q_j, respectively, is calculated using the quotient between the product of the charges and the squared distance from I to J, represented in (2).

$$a_{ij} = ||\overrightarrow{a_{ij}}|| = ||\overrightarrow{a_{ji}}|| = \frac{q_i \cdot q_j}{d_{ij}^2} \tag{2}$$

where $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the Euclidean distance between points I and J.

After calculating the value of the attraction between each pair of points, that is, after constructing the matrix A, it is possible to find the pair with the greatest "admissible attraction." "Admissible" because capacity restrictions must be validated, which means that the sum of the charges of the two points with maximum attraction cannot be greater than the capacity available for the sector. Suppose that Q represents the maximum charge (amount of waste) that can be collected, and I and J are two generic points with charges q_i and q_j, respectively. The objective here is to find the pair (I, J), such that $arg\ max\{a_{ij} \in A : q_i + q_j \leq Q\}$.

After that, the two selected points will join at a new point $C = (x_C, y_C)$, with charge q_C as illustrated in Fig. 2.

Distances k_i and k_j represented in Fig. 2 are equal to $k_i = \frac{q_j}{q_i+q_j} \cdot d_{ij}$ and $k_j = (1 - \frac{q_j}{q_i+q_j}) \cdot d_{ij} = \frac{q_i}{q_i+q_j} \cdot d_{ij}$.

Unless the charges q_i and q_j are exactly the same, point C is not at the same distance between I an J. C is always closer to the point with higher charge.

After the first iteration, the resulting charge usually attains a larger value. This means that, in the second iteration, that charge will be given an unfair preference because of its dimension, to be presented in the next fusion. Therefore, when

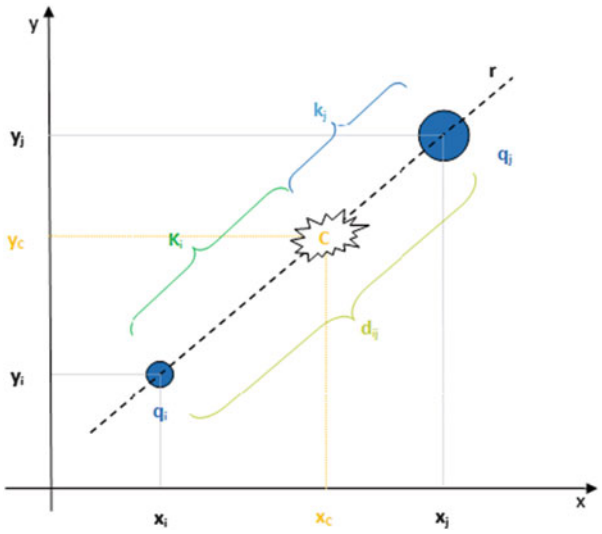


Fig. 2 Union of points I e J in a new point C

equation (2) is used to calculate attractions between charges, the result will be quite unbalanced. Charges will increase until the capacity restriction is not violated.

To deal with this difficulty, that is, the resulting "extreme force" of attraction, a few changes are proposed for the first equation (2). This means that, the exponent of the denominator will increase iteration after iteration. The resulting attraction matrix is A^d, now defined by expression (3):

$$||\overrightarrow{a_{ij}^d}|| = ||\overrightarrow{a_{ji}^d}|| = \frac{q_i \cdot q_j}{d_{ij}^{2+NIP-NS}} \tag{3}$$

NIP represents the Number of Initial Points and NS is the current Number of Sectors. In the beginning, $NS = NIP$ after the first iteration $NS = NIP - 1$, and then $NS = NIP - 2$, and so on, until the desired number of sectors is obtained.

This change in the matrix of attractions not only prevents unbalanced sectors, but also increases the compactness and the contiguity of the resulting sectors.

In the specific case of waste collection, another change was done to "open" the distribution of the set of points to collect. It was convenient to take into account some other points (landfills or transfer stations). These will have fixed charges and exert an attractive force on the resulting charge, which is proportional to the point's receiving capacity.

Figure 3 exemplifies what has been said. Suppose the points with maximum attraction are P_1 and P_2, with charges q_1 and q_2, respectively. P represents the resulting point with charge $q_t = q_1 + q_2$. Suppose also that there are two landfills with different receiving capacities. The capacity of each landfill is represented by

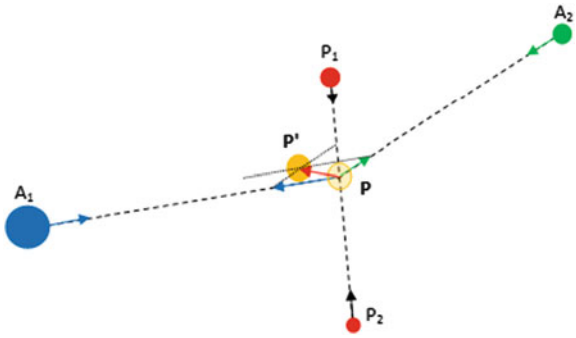


Fig. 3 The action of the resultant force on point P

Table 1 Weekly schedule

Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Circuit	C_1	C_2	C_1	C_2	C_1	C_2

its charge: landfill A_1 with charge qA_1 and A_2 with qA_2 and $qA_1 > qA_2$. Influenced by the resulting force exerted by deposition points, the point P will be displaced to the position P'.

5.1.2 Repulsions for Admissibility

Attractive forces were operated in the previous section. The comparison with electromagnetism will be enhanced by including "repulsive forces". In fact, the following situations only make sense in this framework if repulsive forces are assumed. The first situation refers to the case in which different containers cannot be collected by the same vehicle. Suppose that two containers are close. One of the containers requires a crane to be collected (a large vehicle) and the other is in a narrow street, and therefore a smaller vehicle is mandatory. Even though both containers are close, they cannot be included in the same sector. Another situation is related to the collection frequency.

Generally, different locations in the same municipality require different frequencies of collection. Frequency is usually related to population density, and also to the proximity to public spaces, such as schools or hospitals.

Consider, for instance, a simple situation where two regions A and B must be collected. Suppose that in region A the collection is done every day, and in region B the collection is done three times a week. Sundays are dedicated to regions with daily collection (region A is an example). Table 1 characterizes a weekly work schedule composed of two circuits C_1 and C_2 repeated until the end of the week. This means that the planning is done for two days and repeated three times. Remember that region B must be collected three times a week and, obviously, the collection must be spread throughout the week.

Region A, where waste is collected every day, must belong to circuits C_1 and C_2, and region B will be in C_1 or in C_2. For instance, if region B is included in circuit C_1, then waste will be collected on Mondays, Wednesdays and Fridays.

When planning for two days (sector 1 with circuit C_1 and sector 2 with circuit C_2), an "exact copy" will be made of containers of region A. Therefore, when the process of sectorization is initiated, it is necessary to guarantee that the same container (the original and the copy) will not belong to the same sector. Moreover, a container and its copy must repulse each other, thus preventing the same container to be collected twice in the same day.

5.1.3 Different Levels of Attraction and Repulsion

The dichotomy attraction vs repulsion may become too strict, and for that reason it is necessary to make improvements and adaptations to reality. Imagine another situation comprising two locations separated by a river, a small Euclidian distance – the two locations seem to be close but in terms of the road network, the distance between them may be quite large. Consequently, it is not right to prevent those two locations from belonging to the same sector, but a negative weight should be associated with this "mix".

Other intermediate situations were considered besides the dichotomous *attraction vs repulsion*. The justification to create more levels is not just geographical. Broader situations could be incorporated, such as a decision-maker who might say: "although it is not required, we prefer waste collection in regions A and B to be done on the same day (or on different days)."

When this kind of information (geographical or past experiences) is accompanying pairs of points I and J, a symmetric matrix S (with null diagonal) is generated, $S = [s_{ij}], i, j = 1, \ldots, n$, where n is the number of points in the system,

$$s_{ij} = \begin{cases} -1 & \text{, points } i \text{ and } j \text{ will be in different sectors} \\ -0.5 & \text{, points } i \text{ and } j \text{ have some repulse} \\ 0 & \text{, points } i \text{ and } j \text{ are independent} \\ 0.5 & \text{, points } i \text{ and } j \text{ have some affinity} \\ 1 & \text{, points } i \text{ and } j \text{ have much affinity} \end{cases}$$

If the intention is to add this new information to the matrix of attraction A^d presented before, then the new matrix A' is obtained, as defined in (4).

$$||\overrightarrow{a'_{ij}}|| = ||\overrightarrow{a'_{ji}}|| = \frac{q_i \cdot q_j}{d_{ij}^{2+NIP-NS}} \cdot (s_{ij} + 1) \tag{4}$$

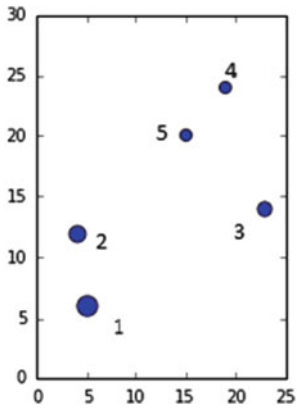


Fig. 4 Five points to form four sectors

Table 2 Position and quantities of the five points

Point	(x, y)	Quantity
1	(5,6)	300
2	(4,12)	200
3	(23,14)	140
4	(19,24)	100
5	(15,20)	100

Example

To illustrate the idea, consider the following example (see Fig. 4) with five points and four sectors to be generated.

In practice, each point represents a container (or group of containers). The position of each point and the quantity to be collected are known, Table 2.

Suppose also that for some reason points 1 and 2 repulse each other ($s_{12} = -1$), and that all other points are independent of each other (if $(i,j) \neq (1,2)$ and $(i,j) \neq (2,1)$ then $s_{ij} = 0$). The quantity in each sector must be not greater than 500. $NIP = 5$ and, in the first iteration, $NS = NIP$. The matrix of attractions A' is

$$A' = \begin{bmatrix} 0 & 0 & 108 & 58 & 123 \\ 0 & 0 & 77 & 54 & 108 \\ 108 & 77 & 0 & 121 & 140 \\ 58 & 54 & 121 & 0 & \mathbf{313} \\ 123 & 108 & 140 & \mathbf{313} & 0 \end{bmatrix}.$$

The application of the proposed sectorization method results in the attraction of points 4 and 5, see Fig. 5, given the quantities, distances and repulsion between points 1 and 2.

⋆

Every time two points are joined in the same sector, the size of matrix A^d decreases and a new matrix must be calculated. After two points are united, say

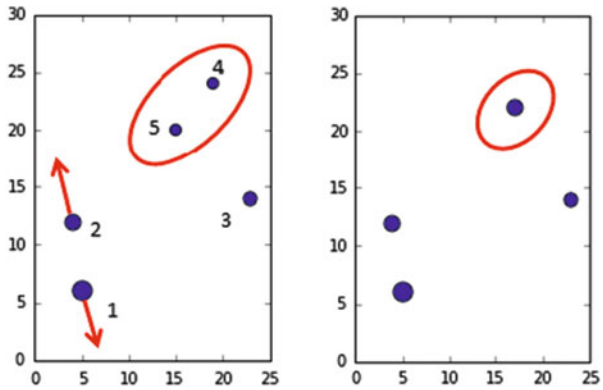


Fig. 5 Repulsion between points 1 and 2 and the resulting attraction between points 4 and 5

Table 3 Δ operation

Δ	-1	-0.5	0	0.5	1
-1	-1	-1	-1	-1	-1
-0.5	-1	-0.5	-0.5	-0.5	-0.5
0	-1	-0.5	0	0.5	1
0.5	-1	-0.5	0.5	0.5	0.5
1	-1	-0.5	1	0.5	1

I and J, the relations between the newly obtained point C and all the others must be defined, taking into account the previous relations between I and J, and the others.

Define the commutative operation $\Delta : M \times M \rightarrow M$, considering $M = \{-1; -0.5; 0; 0.5; 1\}$ and Δ, as expressed in Table 3.

As an example, suppose that point I presents a relation of absolute repulsion relatively to point G, and point J does not attract or repulse point G. In that case, J and G are absolutely independent. When the new point C is created by joining I and J, the resulting relation between C and G will be of repulsion ($s_{cg} = s_{gc} = -1$).

It should be noted that the relations of repulsion should be the exception and not the rule.

5.1.4 Evaluating Sectors

The resulting sectors are evaluated taking into consideration three characteristics described at the beginning of this paper.

Equilibrium

A coefficient of variation in the amount of waste (CV_q) is considered to evaluate the balance. It is calculated as follows, for a group of k sectors with quantities q_i,

$i = 1, \ldots, k$:

$$CV_q = \frac{s'_q}{\overline{q}} \tag{5}$$

where $\overline{q} = \frac{\sum_{i=1}^{k} q_i}{k}$ and $s'_q = \sqrt{\frac{1}{k-1} \cdot \sum_{i=1}^{k} (q_i - \overline{q})^2}$

Hence, in terms of the quantity of waste collected, balanced sectors should have a CV_q as close to zero as possible.

Compactness

The compactness d_i of each sector i, here perceived as a measure of concentration of waste to be collected in each sector, is defined by (6):

$$d_i = \frac{\sum_{j} q_{ij}}{dist(o_i, p_i)} \tag{6}$$

where q_{ij} represents the charge (or amount of waste) of the point j in sector i, and $dist(o_i, p_i)$ is the distance (Euclidean) between the centroid of the sector i, o_i, and the point of the same sector, p_i, that is farthest from o_i.

Higher values of d_i represent higher values of compactness, which means a "higher density" in sector i. This measure does not guarantee *extreme compactness*, but prevents spread out sectors.

In the same sectorization process, it is not desirable to have one sector with a high concentration (compactness), while others present poor values. Therefore, similarly to the balance analysis, compactness is evaluated using the coefficient of variation CV_d defined in (7).

$$CV_d = \frac{s'_d}{\overline{d}} \tag{7}$$

where $\overline{d} = \frac{\sum_{i=1}^{K} d_i}{K}$ and $s'_d = \sqrt{\frac{1}{K-1} \cdot \sum_{i=1}^{K} (d_i - \overline{d})^2}$.

A good sectorization must have a CV_d close to zero.

Contiguity

Consider the original graph $G = (V, E)$, with $|V| = N$, where K sectors must be constructed ($K < N$). The evaluation of the contiguity of the K sectors is calculated using the adjacency matrices obtained from the k subgraphs $G'_i = (V'_i, E'_i)$ ($i = 1, \ldots, K$), where V'_i and E'_i represent the set of vertices and the set of edges of subgraph G'_i, respectively. The number of vertices of each sector i is represented by: $|V'_i| = n_i, i = 1, \ldots, K$.

For each subgraph, G'_i also considers the symmetric matrix given by $M^i = [m^i_{wj}]_{w,j=1,\ldots,n_i}$ with main diagonal zero, where

$$m^i_{wj} = \begin{cases} 1 \text{ if in sector i exists a walk between w and j} \\ 0 \qquad\qquad\qquad\quad \text{otherwise} \end{cases}$$

$$M^i = \begin{bmatrix} 0 & m^i_{12} & m^i_{13} & \ldots & m^i_{1n_i} \\ m^i_{21} & 0 & m^i_{23} & \ldots & m^i_{2n_i} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m^i_{n_i1} & m^i_{n_i2} & m^i_{n_i3} & \ldots & 0 \end{bmatrix}.$$

If for all $j \in \{1, \ldots, n_i\}$, $\sum_{w=1}^{n_i} m^i_{wj} = n_i - 1$ or for all $w \in \{1, \ldots, n_i\}$ the condition $\sum_{j=1}^{n_i} m^i_{wj} = n_i - 1$ is met, which is equivalent, then sector i is contiguous.

The next expression (8) for c_i is used to measure contiguity (c_i) in each sector I:

$$c_i = \frac{\sum_{j=1}^{n_i} (\sum_{w=1}^{n_i} m^i_{wj})}{n_i(n_i - 1)} \tag{8}$$

For every sector i, $0 \leq c_i \leq 1$.

This is not enough to characterize the "level" of contiguity. The quality of the sectors must combine the contiguity of all sectors produced. The weighted average of *isolated contiguities* is used to evaluate the resulting contiguity.

$$\bar{c} = \frac{\sum_{i=1}^{K} c_i \cdot n_i}{N}. \tag{9}$$

$\bar{c}$ is a value that is always between zero and one.

From the perspective of contiguity, a good sectorization must have a $\bar{c}$ value as close to one as possible.

5.2 Routes: Traveling in Sectors

A vehicle leaves the garage, and returns empty at the end of a work day. After the vehicle is filled, it is emptied in special points (landfills or transfer stations) that are available for this purpose. Each vehicle is responsible for collecting waste in each sector. The model for the arc routing problem linked to the case study is presented in [46].

Moreover, some deposition points may have limitations regarding the number of visits received daily. The limitations are mainly due to the small size of the existing facilities. The deposition in different points may represent different costs for the vehicle. Landfills and transfer stations are not simple "crossing points". Each time a vehicle enters a deposition point is emptied.

5.2.1 Arc Routing Model with Limited Multi-landfills

The Mixed Capacitated Arc Routing Problem with Limited Multi-Landfills (MCARP-*LML*) is an MCARP with multiple landfills ([44] and [46]) some of which with a limited number of visits for waste disposal. The objective is to minimize the cost of travels between the landfill/transfer station and the garage or depot, while the demand is met without exceeding the capacity of the vehicle. This model is based on the work by [18] for the MCARP. The main differences between those two models is that in the model presented by [18] landfill and garage are represented by the same point. As a consequence, the landfill is unique and has no limitations regarding the number of empties. In the model used here, the number of landfills (which are distinct from the garage) is greater or equal to one. As previously stated, only the case with one vehicle is addressed. The situation with different vehicles is presented by the authors in [46].

5.2.2 Evaluating Routes

The quality of routes is once again evaluated considering the equilibrium between the elements in the group. In this case, the time spent collecting waste in each sector must be as similar as possible. Considering K routes (K sectors), $r_i, i = 1, \ldots, K$ represents the time to collect sector i, the coefficient of variation CV_r is defined in (10) as:

$$CV_r = \frac{s'_r}{\bar{r}} \tag{10}$$

where $\bar{r} = \frac{\sum_{i=1}^{K} r_i}{K}$ and $s'_r = \sqrt{\frac{1}{K-1} \cdot \sum_{i=1}^{K} (r_i - \bar{r})^2}$.

A good set of routes must have a CV_r close to 0.

6 Computational Results

This section presents the computational results for the real case study described in this paper. The results were obtained using the CPLEX 12.6 (IBM ILOG CPLEX Optimization Studio) solver in an Intel Core i7 2.00 GHz computer with Turbo Boost up to 3.1 GHz computer, and 4.00 GB of RAM. In the most common situation throughout the year, three *freguesias* must be collected every day, and the others twice a week. Two circuits are designed for each day, except Sundays. The week is divided into three parts: the first part includes Monday, Tuesday and Wednesday; the second is a copy of the first and includes Thursday, Friday and Saturday; finally, the third part, which corresponds to Sunday, is only dedicated to the three regions (*freguesias*) with daily demand.

The electromagnetism-based approach was initially applied and the results are depicted in Fig. 6. Euclidean distances between elementary units (*freguesias*) of the municipality of Monção were used.

Table 4 reveals the values obtained for the *three measures of quality*.

A possible schedule is presented in Table 5 considering the three parts of the week characterized before.

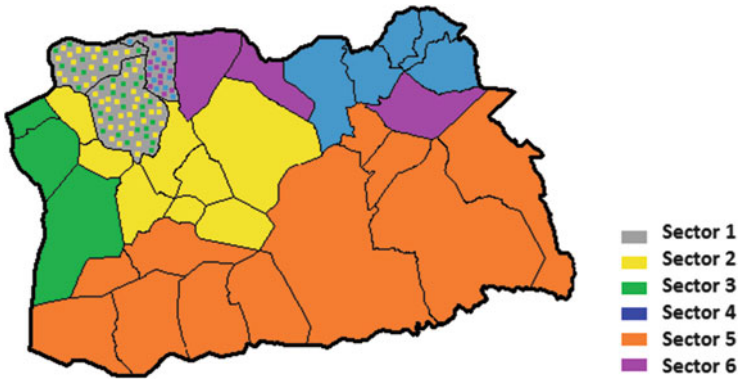


Fig. 6 Different colors identify diverse sectors of the municipality of Monção

Table 4 Measures of sector's *quality*

Equilibrium	Compactness	Contiguity
0.1744	0.2712	0.6667

Table 5 Weekly schedule (S_i represents Sector i)

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
S_1	S_4	S_6	S_1	S_4	S_6	S_1
S_5	S_2	S_3	S_5	S_2	S_3	–

Table 6 Characteristics of the six routes

ID[a]	*#T*	*#V*	*#R*	*#NR*	*Gap* (%)	*Comp.Time* (*sec*)	*OF*
S_1	4	183	145	121	0	92.11	36,111.448
S_2	4	200	165	124	0	2642.08	43,619.712
S_3	3	136	110	82	0	36.23	27,867.984
S_4	4	189	151	153	0	4877.91	46,561.504
S_5	4	206	161	133	0.28	3 h[b]	50,937.096
S_6	4	157	114	126	0.04	3 h[b]	39,376.232

[a]Instances are available in
http://www.inescporto.pt/~amr/Limited_Multi_Landfills/RealCase/1Vehicle/
[b]After 3 h running

One route was determined for each of the six sectors, and only one vehicle was considered. The results are presented in Table 6, where
#T – is the number of trips.
#V – represents the number of vertices.
#R – is the number of required edges (or arcs).
#NR – is the number of non-required edges (or arcs).

The values in the last three columns reflect the *Gap*(%), the computational time in seconds and the objective function (seconds).

When (only) Euclidean distances between *freguesias* are considered, just by looking at the map the real resulting sectors do not seem to be the most suitable. In fact, there are situations where even though two *freguesias* are close, there is no link (or road) between them. Figure 7 shows all the possible direct links (green lines) between two *freguesias*.

The initial signs of attraction/repulsion between two *freguesias* were redefined taking into consideration the reality of road connections and distances. The resulting sectorization is presented in Fig. 8.

The new sectors obtained feature a better equilibrium when the amount of waste and better contiguity are taken into account, as confirmed in Table 7.

The resulting planning is provided in Table 8.

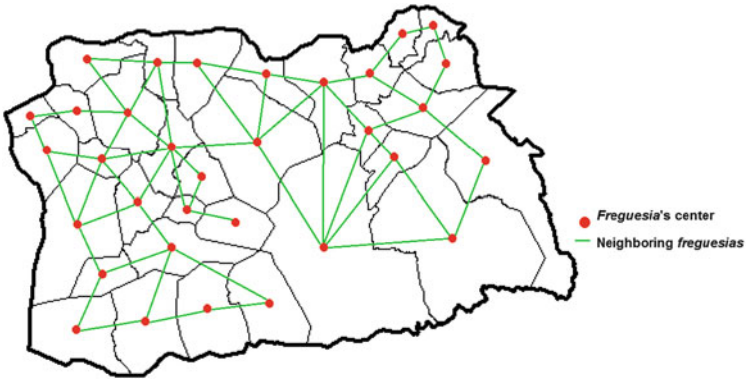


Fig. 7 Linked *freguesias*

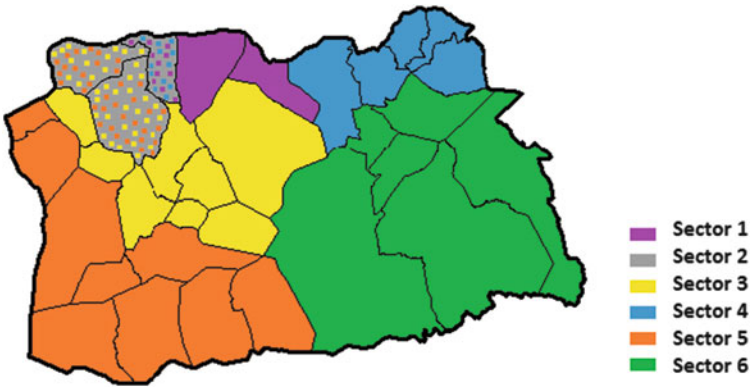


Fig. 8 Different colors identify different sectors of the municipality of Monção (scenario 2)

Table 7 Measures of sector's quality

Equilibrium	Compactness	Contiguity
0.1558	0.3224	0.8130

Table 8 Weekly schedule (S_i represents Sector i)

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
S_1	S_2	S_4	S_1	S_2	S_4	S_2
S_3	S_6	S_5	S_3	S_6	S_5	–

Taking into consideration the new sectors, new routes are calculated and presented in Table 9.

After analyzing and comparing the two scenarios in terms of routes (results displayed in Tables 6 and 9), it was possible to confirm that the variation coefficient decreased from 0.2009 to 0.1038 when the information regarding the topology of the territory was considered.

Table 9 Characteristics of the new six routes (scenario 2)

ID[a]	*#T*	*#V*	*#R*	*#NR*	*Gap* (%)	*Comp.Time* (*sec*)	*OF*
S_1	4	157	106	114	0	582.05	37,401.512
S_2	4	183	145	121	0	92.11	36,111.448
S_3	4	200	165	124	0	2642.08	43,619.712
S_4	4	189	151	153	0	4877.91	46,561.504
S_5	4	207	162	166	3.78	3 h[b]	43,924.176
S_6	3	143	117	89	0	429.70	38,296.704

[a]Instances are available in
http://www.inescporto.pt/~amr/Limited_Multi_Landfills/RealCase/1Vehicle/
[b]After 3 h running

7 Conclusions

This paper deals simultaneously with sectorization and routing problems. It is based on a real-life waste collection problem in Monção, Portugal, but the authors believe that the methods developed can be extended to other applications. That is exactly the case of the new method for sectorization, driven by electromagnetism and Coulomb's Law. The first phase of the approach to the real-life problem was dedicated to sectorization. Not only was the division of the territory considered, but also the frequency with which waste is collected, a critical issue in these types of applications. The new method was able to address the situation by considering geographical information and several quality measures, such as equilibrium, contiguity, and compactness. Additionally, the method allows decision-makers to define levels of "attraction and/or repulsion", meeting their expectations of controlling the results. Special characteristics related to the number and type of deposition points also served as motivation for this work. The second part addresses the routing problems in each sector. The new model *MCARP-LML* – Mixed Capacitated Arc Routing Problem with Limited Multi-Landfills, based on a model from the literature, was proposed and tested in real instances. The results obtained confirmed the effectiveness of the entire approach.

Acknowledgements This work was partially financed by National Funds through the FCT-Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "Project SEROW/ PTDC/ EGE-GES/ 121406/ 2010", and by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by National Funds, through the FCT within "Project NORTE-07-0124-FEDER-000057".

References

1. Amaya, A., Langevin, A., Trépanier, M.: The capacitated arc routing problem with refill points. Les Cahiers du GERAD G-2004-100 (2004)

2. Assad, A.A., Golden, B.L.: Arc routing methods and applications. Handbooks in operations research and management science, vol. 8, pp. 375–483. (1995)
3. Belenguer, J.M., Benavent, E.: The capacitated arc routing problem: valid inequalities and facets. Comput. Optim. Appl. **10**, 165–187 (1998)
4. Benzarti, E., Sahin, E., Dallery, Y.: Operations management applied to home care services: analysis of the districting problem. Decis. Support Syst. **55**, 587–598 (2013)
5. Bertolazzi, P., Bianco, L., Ricciardelli, S.: A method for determining the optimal districting in urban emergency services. Comput. Oper. Res. **4**(1), 1–12 (1977)
6. Bozkaya, B., Erkut, E., Laporte, G.: A tabu search heuristic and adaptive memory procedure for political districting. Eur. J. Oper. Res. **144**, 12–26 (2003)
7. Caro, F., Shirabe, T., Guignard, M., Weintraub, A.: School redistricting: embedding GIS tools with integer programming. J. Oper. **55**, 836–849 (2004)
8. Chu, F., Labadi, N., Prins, C.: A scatter search for the periodic capacitated arc routing problem. Eur. J. Oper. Res. **169**, 586–605 (2006)
9. D'Amico, S.J., Wang, S.-J., Batta, R., Rump, C.M.: A simulated annealing approach to police district design. Comput. Oper. Res. **29**(6), 667–684 (2002)
10. Dror, M. (ed.): Arc Routing: Theory, Solutions and Applications. Springer Science & Business Media (2012)
11. Eiselt, H.A., Gendreau, M., Laporte, G.: Arc routing problems (part I): the Chinese postman problem. Oper. Res. **43**(2), 231–242 (1995)
12. Eiselt, H.A., Gendreau, M., Laporte, G.: Arc routing problems (part II): the rural postman problem. Oper. Res. **43**(3), 399–414 (1995)
13. Fleury, G., Lacomme, P., Prins, C.: Evolutionary Algorithms for Stochastic Arc Routing Problems. In: Raidl, G.R., et al. (eds.) Applications of Evolutionary Computing. Lecture Notes in Computer Science, vol. 3005, pp. 501–512. Springer, Berlin (2004)
14. Garfinkel, R.S., Nemhauser, G.L.: Optimal political districting by implicit enumeration techniques. Manag. Sci. **16**(8), 495–508 (1970)
15. Geroliminis, N., Kepaptsoglou, K., Karlaftis, M.G.: A hybrid hypercube – genetic algorithm approach for deploying many emergency response mobile units in an urban network. Eur. J. Oper. Res. **210**, 287–300 (2011)
16. Ghiani, G., Improta, G., Laporte, G.: The capacitated arc routing problem with intermediate facilities. Networks **37**(3), 134–143 (2001)
17. Golden, B., Wong, R.: Capacitated arc routing problem. Networks **11**, 305–315 (1981)
18. Gouveia, L., Mourão, M.C., Pinto, L.S.: Lower bounds for the mixed capacitated arc routing problem. Comput. Oper. Res. **37**, 692–699 (2010)
19. Hanafi, S., Freville, A., Vaca, P.: Municipal solid waste collection: an effective data structure for solving the sectorization problem with local search methods. INFOR **37**(3), 236–254 (1999)
20. Hess, S.W., Samuels, S.A.: Experiences with a sales districting model: criteria and implementation. Manag. Sci. **18**(4), 41–54 (1971)
21. Hess, S.W., Weaver, J., Siegfeldt, H., Whelan, J., Zitlau, P.: Nonpartisan political redistricting by computer. Oper. Res. **13**(6), 998 (1965)
22. Iannoni, A.P., Morabito, R., Saydam, C.: An optimization approach for ambulance location and the districting of the response segments on highways. Eur. J. Oper. Res. **195**, 528–542 (2009)
23. Kalcsics, J., Nickel, S., Schröder, M.: Towards a unified territorial design approach – applications, algorithms and GIS integration. Top **13**(1), 1–56 (2005)
24. Kip, A.F.: Fundamentals of Electricity and Magnetism. McGraw-Hill Series in Fundamentals of Physics: An Undergraduate Textbook Program. McGraw-Hill, New York (1969)
25. Lacomme, P., Prins, C., Ramdane-Chérif, W.: Evolutionary Algorithms for Multiperiod Arc Routing Problems. In: Proc. of the 9th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based systems (IPMU 2002). (2002)
26. Lacomme, P., Prins, C., Sevaux, M.: Multiobjective Capacitated Arc Routing Problem. In: Evolutionary Multi-Criterion Optimization. Springer, Berlin, Heidelberg (2003)
27. Lacomme, P., Prins, C., Ramdane-Chérif, W.: Competitive memetic algorithms for arc routing problems. Ann. Oper. Res. **131**, 159–185 (2004)

28. Lacomme, P., Prins, C., Sevaux, M.: A genetic algorithm for a bi-objective capacitated arc routing problem. Comput. Oper. Res. **33**, 3473–3493 (2006)
29. Lamata, M., Pcláez, J., Sierra, J., Bravo, J.: A sectoring genetic algorithm for the urban waste collection problem. In: Reusch, B. (ed.) Computational Intelligence. Volume 1625 of Lecture Notes in Computer Science, pp. 642–646. Springer, Berlin/Heidelberg (1999)
30. Letchford, A.N., Oukil, A.: Exploiting sparsity in pricing routines for the capacitated arc routing problem. Comput. Oper. Res. **36**(7), 2320–2327 (2009)
31. Levy, L., Bodin, L.: The arc oriented location routing problem. INFOR **27**(1), 74–94 (1989)
32. Liu, T., Jiang, Z., Chen, F., Liu, R., Liu, S.: Combined Location-Arc Routing Problems: A Survey and Suggestions for Future Research. Service Operations and Logistics, and Informatics. IEEE/SOLI 2008. IEEE International Conference on. vol. 2. IEEE (2008)
33. Longo, H., Aragão, M.P., Uchoa, E.: Solving capacitated arc routing problems a transformation to the CVRP. Comput. Oper. Res. **33**, 1823–1837 (2006)
34. Lunday, B.J., Sherali, H.D., Lunday, K.E.: The coastal seaspace patrol sector design and allocation problem. Comput. Manag. Sci. **9**, 483–514 (2012)
35. Male, J.W., Liebman, J.C.: Districting and routing for solid waste collection. J. Environ. Eng. Div. **104**(1), 1–14 (1978)
36. Marianov, V., Fresard, F.: A procedure for the strategic planning of locations, capacities and districting of jails: application to Chile. J. Oper. Res. Soc. **56**, 244–251 (2005)
37. Minciardi, R., Puliafito, P., Zoppoli, R.: A districting procedure for social organizations. Eur. J. Oper. Res. **8**(1), 47–57 (1981)
38. Moreira, L., Oliveira, J., Gomes, A., Ferreira, J.S.: Heuristics for a dynamic rural postman problem. Comput. Oper. Res. **34**(11), 3281–3294 (2007)
39. Mourão, M.C., Nunes, A.C., Prins, C.: Heuristic methods for the sectoring arc routing problem. Eur. J. Oper. Res. **196**(3), 856–868 (2009)
40. Orloff, C.S.: A fundamental problem in vehicle routing. Networks **4**, 35–64 (1974)
41. Pezzella, F., Bonanno, R., Nicoletti, B.: A system approach to the optimal health-care districting. Eur. J. Oper. Res. **8**(2), 139–146 (1981)
42. Prins, C., Prodhon, C., Calvo, R.W.: A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In: Lecture Notes in Computer Science, vol. 3906. Springer, Berlin/Heidelberg (2006)
43. Ricca, F., Scozzari, A., Simeone, B.: Political districting: from classical models to recent approaches. Ann. Oper. Res. **204**, 271–299 (2013)
44. Rodrigues, A.M.: Sectores e Rotas na Recolha de Resíduos Sólidos Urbanos. Ph.D. thesis, Faculdade de Engenharia da Universidade do Porto (2014)
45. Rodrigues, A.M., Ferreira, J.S.: Cutting path as a rural postman problem: solutions by memetic algorithms. Int. J. Comb. Optim. Probl. Inform. **3**(1), 22–37 (2012)
46. Rodrigues, A.M., Ferreira, J.S.: Waste collection routing – limited multiple landfills and heterogeneous fleet. Networks (2015). doi:10.1002/net.21597
47. Skiera, B., Albers, S.: Costa: contribution optimizing sales territory alignment. Mark. Sci. **17**, 196–213 (1998)
48. Tagmouti, M., Gendreau, M., Potvin, J.-Y.: Arc routing problems with time-dependent service costs. Eur. J. Oper. Res. **181**, 30–39 (2007)
49. Takashi, T., Yukio, S.: Evaluation of school family system using GIS. Geogr. Rev. Jpn. **76**(10), 743–758 (2003)
50. Wøhlk, S.: A decade of capacitated arc routing. In: Golden, B., Raghavan, S., Wasil, E. (eds). The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces, Springer, pp. 29–48 (2008)
51. Zoltners, A.A., Sinha, P.: Sales territory alignment: a review and model. Manag. Sci **29**(11), 1237–1256 (1983)

Solving Multilocal Optimization Problems with Parallel Stretched Simulated Annealing

José Rufino and Ana I. Pereira

Abstract This work explores the use of parallel computing to solve multilocal optimization problems with Stretched Simulated Annealing (SSA), a method that combines simulated annealing with a stretching function technique. Several approaches to the parallelization of SSA are explored, based on different strategies for the refinement of the initial feasible region in subregions and its allocation to the processors involved. The parallel approaches, collectively named as PSSA (Parallel SSA), make viable what would otherwise be unfeasible with traditional sequential computing: an efficient search of the subregions that allows to find many more optima in a reasonable amount of time. To prove the merits of PSSA, several experimental metrics and numerical results are presented for a set of benchmark problems.

1 Introduction

A multilocal programming problem aims to find all local solutions of the problem

$$\min_{x \in X} f(x), \tag{1}$$

where $f : I\!R^n \rightarrow I\!R$ is a given multimodal objective function and X is a compact set defined by $X = \{x \in I\!R^n : a_i \leq x_i \leq b_i, i = 1, \ldots, n\}$.

J. Rufino (✉)
Polytechnic Institute of Bragança, Bragança, Portugal

Laboratory of Instrumentation and Experimental Particle Physics, University of Minho, Guimaraes, Portugal
e-mail: rufino@ipb.pt

A.I. Pereira
Polytechnic Institute of Bragança, Bragança, Portugal

Algoritmi R&D Centre, University of Minho, Guimaraes, Portugal
e-mail: apereira@ipb.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_21

So, the purpose is to find all local solutions $x^* \in X$ such that

$$\forall x \in V_\epsilon(x^*),\ f(x^*) \leq f(x), \tag{2}$$

for a positive value ϵ.

These problems appear in practical situations like ride comfort optimization [2], Chemical Engineering (process synthesis, design and control [3]), and reduction methods for solving semi-infinite programming problems [10, 18].

The most common methods for solving multilocal optimization problems are based on evolutionary algorithms, such as genetic algorithms [1] and particle swarm algorithms [13]. Additional contributions may be found in [9, 21, 25, 26].

Stretched Simulated Annealing (SSA) was also proposed [14, 16, 17] as a method to solve multilocal programming problems. SSA combines simulated annealing with a stretching function technique in order to identify the local minimizers. However, the numerical results obtained with SSA indicate that, in some tested problems, the method is only able to find a small number of minimizers.

In previous work [20], Parallel Stretched Simulated Annealing (PSSA) was introduced as a first attempt to the parallelization of SSA, based on the refinement of the search domain (feasible region) in a fixed number of equal-width subdomains (subregions), and its deterministic allocation to the processors involved. This first approach, hereafter renamed as PSSA_HoS (PSSA with **Ho**mogeneous decomposition and **S**tatic distribution), proved to be an effective and scalable way to improve the number of optima found in a bounded time.

In this paper the research on PSSA is consolidated by exploring two additional parallelization strategies: (i) PSSA with **Ho**mogeneous decomposition and **D**ynamic distribution (PSSA_HoD), a straightforward alternative to PSSA_HoS in which the assignment of subdomains to processors happens on-demand, thus aiming at better load balancing; (ii) PSSA with **He**terogeneous decomposition and **D**ynamic distribution (PSSA_HeD), a logical successor to PSSA_HoD, based on a recursive adaptive refinement of the feasible region, whose main purpose is to further increase the numerical efficiency of the search process. New filtering criteria that eliminate the false optima produced by the PSSA approaches are also presented.

All PSSA approaches are evaluated and compared, using a set of well known problems. Comparison metrics include search times, parallel search speedups, number of subdomains searched, maximum recursive search depth and number of optima found. A linear metric is introduced to assist the choice of the approach that best suits the desired compromise between search time and number of optima found.

The rest of the paper is organized as follows: Sect. 2 describes the original SSA method; Sect. 3 elaborates on the various PSSA approaches and the filtering criteria used to eliminate false optima; Sect. 4 presents the optimization problems evaluated, and the computing and numerical results of the evaluation experiments; the last section presents some conclusions and directions for future work.

2 Stretched Simulated Annealing

The Stretched Simulated Annealing (SSA) method solves a sequence of global optimization problems in order to compute the local solutions of the minimization problem (1) that satisfy the condition (2). The objective function of each global optimization problem is obtained by applying a stretching function technique [12].

Let x_j^* be a particular solution of problem (1). The mathematical formulation of the global optimization problem is as follows:

$$\min_{a \le x \le b} \Phi_l(x) \equiv \begin{cases} \hat{\phi}(x) \text{ if } x \in V_{\varepsilon^j}(x_j^*),\ j \in \{1, \ldots, N\} \\ f(x) \quad \text{otherwise} \end{cases} \tag{3}$$

where $V_{\varepsilon^j}(x_j^*)$ represents the neighborhood of the optimum solution x_j^* with a ray ε^j, and N is the number of minimizers already detected.

The $\hat{\phi}(x)$ function is defined as

$$\hat{\phi}(x) = \bar{\phi}(x) + \frac{\delta_2[\text{sign}(f(x) - f(x_j^*)) + 1]}{2\tanh(\kappa(\bar{\phi}(x) - \bar{\phi}(x_j^*)))} \tag{4}$$

and

$$\bar{\phi}(x) = f(x) + \frac{\delta_1}{2}\|x - x_j^*\|[\text{sign}(f(x) - f(x_j^*)) + 1] \tag{5}$$

where δ_1, δ_2 and κ are positive constants.

To solve the global optimization problem (3), the simulated annealing (SA) method is used [7]. The SSA algorithm stops when no new optimum is identified after l consecutive runs. For more details see [14, 15].

3 Parallel Stretched Simulated Annealing (PSSA)

The search for optima of nonlinear optimization functions with SSA is easily parallelizable. SSA searches for solutions in a given domain by applying a stochastic algorithm to a specific search domain (feasible region). This procedure is repeated l consecutive times. Augmenting l increases the hit rate of the algorithm, but also increases execution times. An alternative to ameliorate the hit rate is to keep l constant but expand the search effort by splitting the initial search domain in several mutually exclusive subdomains where SSA will be applied independently, either sequentially (one subdomain at a time) or in parallel (many subdomains at the same time).

Running SSA on a subdomain is expected to take no more time (on average) than running it on the entire initial feasible region (in fact, it should be faster, once

subdomains are smaller search regions). Also, applying SSA to one subdomain hasn't any data or functional dependency on any other subdomain, thus easing parallelization.

As such, provided enough processors/CPU-cores, all subdomains could be searched simultaneously, by having one SSA instance running per processor and searching a specific subdomain. Moreover, the overall search time would still be approximately the same as the time spent to conduct a single search on the initial feasible region.[1] On the other hand, if domain decomposition is too fine with relation to the number of available processors, there will be some serialization of the subdomains searches, leading to an increase of the overall search time (such increase depends on the discrepancy between the number of processors and the number of subdomains).

In short, the previous discussion makes clear that a Data Decomposition strategy is applied on the problem domain, and that a Single Program Multiple Data (SPMD) execution model (i.e., several instances of the same SSA implementation, dealing with different subdomains) is particularly suitable to the parallelization of SSA. All PSSA variants discussed in this paper share these general properties. However, they diverge on the way in which subdomains are defined (*homogeneous* vs *heterogeneous* decomposition with respect to subdomain size), as well as in the way in which subdomains are assigned to processors (*static* vs *dynamic* distribution).

3.1 Implementation Details

SSA was originally implemented in C [8], under Microsoft Windows®. The original code was first ported to Linux and then used as the basis for the development of the PSSA variants. The decision to use the Linux environment to host the development and execution of PSSA allowed us to exploit local parallel computing facilities (built on Linux) and is also in line with the fact that Linux clusters are the dominant environment for the execution of parallel scientific applications [19, 22, 24].

In order to allow PSSA to execute transparently, both on multi-core shared memory systems, and on distributed memory HPC clusters, PSSA was built on top of MPICH2 [6], an implementation of the Message Passing Interface (MPI) specification [11] for parallel applications that follow the Message Passing paradigm.

In this context, all PSSA variants operate in the same basic MPI configuration: *slave* MPI processes apply SSA to problem subdomains; a *master* process performs pre-processing, coordination and post-processing tasks; if c CPU-cores are enrolled in the execution of PSSA, one core is reserved for the *master* and the remaining $c - 1$ cores are for *slaves*, with one *slave* per core (this is the process mapping that better exploits the available parallelism of our execution environment).

[1] Ignoring the time to spawn and coordinate all SSA instances, and post-process results.

In PSSA the overall number of *slaves* is independent of the overall number of subdomains. This is both necessary and convenient: if the number of *slaves* were to always match the number of subdomains then, with fine-grain decompositions, a one-to-one mapping of *slaves* to CPU-cores could be unfeasible, preventing efficient PSSA executions. Thus, by separating the definition of the number of *slaves* from the definition of the number of subdomains, each quantity can be tuned at will.

The way in which the initial problem domain is decomposed and *slaves* get subdomains assigned depends on the PSSA variant: the *master* may be the one responsible for the partitioning of the problem domain and assignment of subdomains to *slaves*, or *slaves* may conduct themselves such tasks autonomously; in all cases the *master* is responsible for a final post-processing phase in which all optima candidates found by *slaves* are filtered using the criteria described in Sect. 3.5.

The final filtering of optima candidates should be an efficient process: depending on the specific optimization problem, it may have to cope with candidates in the order of thousands or even millions, that must be stored in efficient containers. Because ANSI C has no built-in container data types (e.g., lists, sets, etc.) it was necessary to reuse an external implementation for that purpose. At the same time, dependencies on additional libraries were minimized, so that PSSA could be easily compiled on any Linux system (provided MPICH2 was available). The choice was to use the GLIBC `tsearch` function family [23], that provides a very efficient implementation of balanced trees (more precisely, of Red-Black-Trees [4]).

All PSSA variants are able to save (if requested) the optima candidates in CSV raw files. These raw files may be later re-filtered, using the same criteria or newest/refined ones, thus avoiding the need to repeat (possibly lengthy) PSSA executions. Moreover, the final optima that result from the filtering process, may also be saved in CSV files that follow the same format of optima candidates raw files.

3.2 Homogeneous Decomposition, Static Distribution (PSSA_HoS)

This section provides an updated description of the first approach to the parallelization of SSA [20]. This approach, now rechristened as PSSA_HoS, laid the foundation for the novel approaches introduced in this paper (PSSA_HoD and PSSA_HeD).

The initial search domain of an optimization function is the feasible region defined by an n-dimensional interval, I, given by the cartesian product of n intervals, one per each problem dimension: $I = I_1 \times I_2 \times \ldots \times I_n$. An *homogeneous decomposition* of I subdivides each initial interval I_i (with $i = 1, 2, \ldots, n$) in 2^m subintervals, such that each subinterval has the same granularity g (relative width), as given by

$$g = \frac{1}{2^m}, \text{ with } m \in \mathbb{N}_0 \tag{6}$$

Table 1 Decomposition granularity (g) and number of initial subdomains (s_{init})

m	g	s_{init} with $n = 2$	s_{init} with $n = 3$
0	1.0	1	1
1	0.5	4	8
2	0.25	16	64
3	0.125	64	512
4	0.0625	256	4096
5	0.03125	1024	32,768

A subdomain is thus a particular combination of subintervals (one subinterval per problem dimension). It then follows that the overall number of initial subdomains with granularity g, that is generated for n dimension problems, is given by

$$s_{init} = \left(\frac{1}{g}\right)^n = 2^{m \times n} \tag{7}$$

For instance, if the feasible region of a 2-dimensional ($n = 2$) function $f(x, y)$ is $I = [-10.0, 10.0] \times [-5.0, 5.0]$, the *homogeneous decomposition* of this initial search domain with $m = 2$ (or $g = 0.5$) originates $s_{init} = 4$ subdomains: $[-10.0, 0.0] \times [-5.0, 0.0]$, $[-10.0, 0.0] \times [0.0, 5.0]$, $[0.0, 10.0] \times [-5.0, 0.0]$, $[0.0, 10.0] \times [0.0, 5.0]$.

Table 1 shows the number of initial subdomains (s_{init}) as a function of the number of dimensions (n) and the granularities (g) used for the study presented in this paper.

In PSSA_HoS, the initial *homogeneous decomposition* is the only decomposition performed (no further refinements take place) and so the overall number of subdomains processed, s, is such that $s = s_{init}$. The homogenous subdomains generated by this decomposition are then assigned to the MPI *slave* processes as described next.

Once *homogeneous decomposition* is a regular decomposition, *slave* processes may easily select, in a completely autonomous and deterministic way, a specific set of subdomains, without any *master* intervention. In this sense, the distribution of subdomains among *slaves* is in fact a self-assignment, whose result is always the same for the same number of subdomains and *slaves*; it may thus be classified as a *static distribution*. The specific way in which this is accomplished is as follows: let c be the overall number of MPI processes (1 *master* and $c - 1$ *slaves*), so that the MPI rank (specific MPI process identifier) of the *master* is $r = 0$ and the ranks of the *slaves* are $r = 1, 2, \ldots, c - 1$; each *slave* generates all s possible subdomains, such that each subdomain has a specific index $i = 0, 1, \ldots, s - 1$; a *slave* r will then claim all subdomains i for which $[i \mod (c - 1)] + 1 = r$ (where "mod" is the remainder of the integer division). This ensures that each *slave* chooses approximately the same number of subdomains (in the worst case, when $[s \ mod \ (c - 1)] \neq 0$, there will be some *slaves* with an excess of one subdomain to handle). It also ensures that subdomains are assigned rotatively among *slaves*, which usually promotes uniform load balancing (otherwise, a *slave* may finish too

soon/too late because its subdomains belong to a zone of the feasible region where SSA executes faster/slower).

While *slaves* are working, the *master* sits idle, waiting for the sets of optima candidates found by the *slaves*. The *slaves* accumulate their optima candidates and only at the end are they sent to the *master* for post-processing filtering that will use the criteria defined in Sect. 3.5. Therefore, the only communication between *master* and *slaves* takes place at the end of the computation, and so the computing power of the CPU-cores/cluster nodes that participate in PSSA_HoS is fully exploited.

This distribution only makes sense if all *slaves* have equal computing capability available; such is the case when executing PSSA_HoS in a single dedicated multi-core system or when using an homogeneous subset of nodes from an HPC cluster; if the involved CPU-cores or nodes are heterogeneous, load imbalances will occur.

3.3 Homogeneous Decomposition, Dynamic Distribution (PSSA_HoD)

In spite of the alternate nature of the static distribution and the assumption of homogeneous processors, the stochastic behavior of SSA searches may make PSSA_HoS still exhibit load imbalances: there may be situations where some *slaves* become idle too soon, while others are still busy trying to exhaust their own subdomain set.

The PSSA_HoD approach tries to solve this problem by replacing the *static distribution* of subdomains with a *dynamic distribution*, in which the (still homogeneous) subdomains are exclusively assigned by the *master* to the *slaves*, and such assignment happens on-demand, that is, *slaves* must explicitly request subdomains from the *master*. Assuming that only one subdomain is granted per each request, this prevents the situation where there are still subdomains to process and, at the same time, there are already idle *slaves*. The processors utilization is thus maximized.

On the other hand, PSSA_HoD increases communication traffic: for each subdomain, there will be a *slave* request and a *master* reply (like in PSSA_HoS, *slaves* still accumulate their optima candidates and only at the end are they sent to the *master*). This additional traffic is usually the price to pay when trying to achieve better workload balancing. Trying to figure out, a priori, if such translates in final lower execution times, is a difficult theoretical exercise. Instead, such question is best answered through experimentation. As later revealed by the experiments (see Sect. 4.3.1), PSSA_HoD is indeed (marginally) faster than PSSA_HoS, and such holds mainly for small granularities/high subdomain counts (i.e., when finer load balancing becomes possible, and thus load balancing becomes more effective).

3.4 Heterogeneous Decomposition, Dynamic Distribution (PSSA_HeD)

PSSA_HoS and PSSA_HoD assume a homogeneous decomposition of the feasible region and that such decomposition is performed only once (whether by the *master* or by *slaves*). The granularity of the search is thus fixed at the very beginning of the search process. However, because finer granularities are expected to return more candidates, choosing very small granularities seems to be the right decision when using PSSA_HoS or PSSA_HoD. On the other hand, a too fine granularity will not only increase the search time as will also increase the number of false optima. A single and definitive value for the granularity may thus be undesirable. In this context, PSSA_HeD offers an alternative approach, by adopting a dynamic and adaptive generation of new subdomains based on the results of the ones scanned so far.

With PSSA_HeD, the feasible region is firstly refined by following the same *homogeneous decomposition* as applied in the PSSA_HoS and PSSA_HoD approaches. The resulting subdomains are searched with SSA and then, if true (i.e., properly verified) optima are found within, further child subdomains will be generated around those optima. Because an optimum may be located anywhere in its hosting subdomain, child subdomains will not only be smaller than the parent, but will also typically vary in size, thus leading to an *heterogeneous decomposition*. The new child subdomains will, in turn, be searched using SSA and, if optima are found, more subdomains will be generated, until a stop criteria is met. Thus, the decomposition is also dynamic and adaptive, and the generated subdomains may be seen as part of an expanding search tree where each node/leaf subdomain refines its ancestor.

A certain subdomain v generates child subdomains v' if three conditions are met:

1. at least one real optima is found in v;
2. the branch of the search tree to which v belongs has not yet achieved a maximum depth or height $h \in \mathbb{N}_0$ (a parameter of PSSA_HeD);
3. all intervals of a child subdomain v' must have a minimum distance of μ (another parameter of PSSA_HeD) from its "parent optimum" (located in v).

A recursive search branch may thus progress as far as $h-1$ levels bellow the root level of the search tree. Thus, when $h = 1$, the search will be confined to the root of the search tree. An infinite search depth, conveyed by $h = 0$, translates in the condition (2) above not being used to control the generation of child subdomains.

In PSSA_HeD, the root of the so-called search tree is made of the initial search (sub)domain(s), and it is also configurable; it may be a single region, in which case its the original feasible region of the optimization problem; but it may also be the set of subdomains that results from the homogeneous decomposition of the given feasible region; it then turns out that the granularity parameter g, used

by PSSA_HoS and PSSA_HoD, is also used by PSSA_HeD, to define the root (sub)domain(s).

Having $g = 1.0$ implies that the root of the search tree matches the original feasible region of the optimization problem. Setting $g < 1.0$ makes the search to start with a grid of homogeneous subdomains; the purpose of this is to increase the probability of finding many optima in the first level of the search tree and thus trigger the generation of many additional new subdomains (otherwise, with $g = 1.0$, the number of optima found will typically be very limited, and their descendant subdomains will be too few and too large to trigger a sustained recursive search process).

With regard to the way in which subdomains are assigned to the MPI *slave* processes, PSSA_HeD is similar to PSSA_HoD in the sense that both use *dynamic distribution* and the *master* process plays a key role in the assignment. There is, however, a fundamental difference between both approaches: in PSSA_HoD, *slaves* pull subdomains from the *master*; in PSSA_HeD, the *master* pushes subdomains to the *slaves*. The reason for the different behavior of PSSA_HeD is this: there may be times when all available subdomains are being processed by *slaves*; in this scenario, if an idle *slave* asked for a subdomain, it would receive none; but that wouldn't mean that the *slave* could terminate; this is because, in the near future, newer unprocessed subdomains might become available, as a byproduct of the current working *slaves*; thus, it is better for the *master* to push subdomains to the *slaves* (when they become available), than the *slaves* asking for them (at the risk of none being available).

In order to achieve this behaviour, the *master* manages a work-queue with all subdomains yet to process, and a slave-status array with the current status (idle/busy) of each *slave*. Initially, the work-queue is populated with the starting grid of homogeneous subdomains and all *slaves* are marked as idle in the slave-status array.

The distribution of subdomains by the *slaves* is then just a matter of iterating trough the slave-status array and, for each idle *slave*, dequeue and send to the *slave* a subdomain from the work-queue, and mark the *slave* as busy. During this iteration, the *master* may find all *slaves* are busy, in which case nothing is removed from the work-queue; it may also find the work-queue to be empty, in which case nothing is assignable to the possible idle *slaves*; however, if the work-queue is empty and if all *slaves* are idle, that means that the overall recursive search process came to an end.

After a distribution round, and assuming the overall search process hasn't finished, the *master* will block, waiting for a message from a *slave*. That message will be empty if the *slave* found no optima in its assigned subdomain; otherwise, it will carry a set of optima found by the *slave* (and already validated by it); in the later case, the optima are added to the optima set of solutions that is being assembled by the *master* (based on all the *slaves* contributions); they are also used to generate new subdomains that will be added to the work-queue; in any case, the *slave* is marked as idle in the slave-status array. The *master* then performs the next distribution round.

3.5 Filtering Criteria

All PSSA variants produce false optima (e.g., the points in the limits of the generated subdomains). This section presents three criteria that, used in sequence, eliminate false optima. In PSSA_HeD they are applied in the *slaves*, right after running SSA in a subdomain; thus, the *master* only receives sets of validated optima. In PSSA_HoS and PSSA_HoD they are applied by the *master* in a post-processing phase.

3.5.1 Criterion 1

At a given moment, there are a total of s subdomains (with $s \geq s_{init}$). Each subdomain v is defined by n intervals, with left and right limits a_i^v and b_i^v, respectively (for $i = 1, \ldots, n$). Consider x^v (with coordinates x_i^v, for $i = 1, \ldots, n$) a minimum found at subdomain v. Define the vector d with components d_i as

$$d_i = \min\left\{\left|a_i^v - x_i^v\right|, \left|x_i^v - b_i^v\right|\right\}, \text{ with } i = 1, \ldots, n$$

and define Δ_1 as

$$\Delta_1 = \sqrt{\sum_{i=1}^{n} d_i^2}, \text{ with } i = 1, \ldots, n.$$

Criterion 1 is then defined as follows:

1. Consider ϵ_1 a positive constant.
2. If $\Delta_1 < \epsilon_1$ then x^v is not a candidate to a minimum of problem (1).

The situation tackled by this criterion is the one in which a subdomain v doesn't have minimum values except in its interval limits.

3.5.2 Criterion 2

Consider the unit vector, 1_i, with all components null except the component i with unit value. Consider the vector e, with component e_i defined as

$$e_i = \frac{f(x^v + \delta 1_i) - f(x^v)}{\delta}, \text{ for } i = 1, \ldots, n$$

with δ being a small positive value. Define also Δ_2 as

$$\Delta_2 = \sqrt{\sum_{i=1}^{n} e_i^2}.$$

Criterion 2 is thus defined as:

1. Consider x^v that satisfy the Criterion 1.
2. If $\Delta_2 > \epsilon_2$ then x^v is not a candidate to a minimum of problem (1).

3.5.3 Criterion 3

Consider the set $X^* = \left\{x^j, j = 1 \ldots, n^*\right\}$ of all solutions that satisfy the Criterion 2 and let n^* be the cardinality of the set X^*.

Criterion 3 is defined as follows:

1. Consider $x^i \in X^*$.
2. The points x^i is a possible minimum value of problem (1) if

$$\left\|x^i - x^j\right\| > \epsilon_3, \text{ for all } j = 1, \ldots, n^* \text{ and } j \neq i$$

4 Evaluation

4.1 Setup

Evaluation was performed in a small commodity HPC cluster of 4 nodes (with one Intel Q9650 3.0GHz quad-core CPU per each node), running Linux ROCKS version 5.4, with the Gnu C Compiler (GCC) version 4.1.2 and MPICH2 version 1.4.

All PSSA executions spawned 16 MPI processes (1 *master* and 15 *slaves*, one MPI process per cluster core), even if there were a surplus of unused *slaves* in certain scenarios. The MPICH2 "machinefile" used was designed to place the first 4 MPI processes (the *master* and the first 3 *slaves*) in a single node and scatter (alternately) the remaining 12 *slaves* across the other 3 nodes. This particular configuration maximizes performance, both for scenarios with very few subdomains (mostly handled by the *slaves* of the 1st node with minimum (or none) network exchanges), and scenarios with lots of subdomains (requiring *slaves* from all the nodes, in which case network exchanges benefit from the dispersion of their endpoints).

Some important parameters used were $\delta = 5.0$ and $l = 5$ for SSA, and $\mu = 0.001$ for PSSA_HeD. For the filtering criteria, $\epsilon_1 = 10^{-4}$, $\epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-2}$.

4.2 Optimization Problems

Four well known multimodal functions were considered: Hartman, Rastrigin, Griewank and Ackley [5]. These were selected because of their large number of local

optima, thus allowing to better assess the numerical performance of the various PSSA approaches. The main characteristics of these problems are presented below.

4.2.1 Hartman Function

The Hartman optimization problem is defined, for $n = 3$, as

$$\min h(x) \equiv -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{3} A_{ij}\left(x_j - P_{ij}\right)^2}$$

where $x \in [0,1]^3$. The matrices A, P and the vector c are defined as follows:

A			P			c
3.0	10	30	0.36890	0.11700	0.26730	1
0.1	10	35	0.46990	0.43870	0.74700	1.2
3.0	10	30	0.10910	0.87320	0.55470	3
0.1	10	35	0.03815	0.57430	0.88280	3.2

This optimization problem has four local solutions and one global solution at $x^* = (0.114614, 0.555649, 0.852547)$ with optimum value $h(x^*) = -3.862782$.

4.2.2 Rastrigin Function

The Rastrigin optimization problem can be defined, for a fixed n, as

$$\min r(x) \equiv 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$$

where $x \in [-5.12, 5.12]^n$. This problem has one global solution defined at $x^* = (0, 0, \ldots, 0)$ with value $r(x^*) = 0$. For $n = 2$, the problem has 121 local solutions. The solutions may be seen in Fig. 1 (the figure represents the graph of $-r(x)$).

4.2.3 Griewank Function

The Griewank optimization problem can be defined, for a fixed n, as

$$\min g(x) \equiv \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{1}}\right) + 1$$

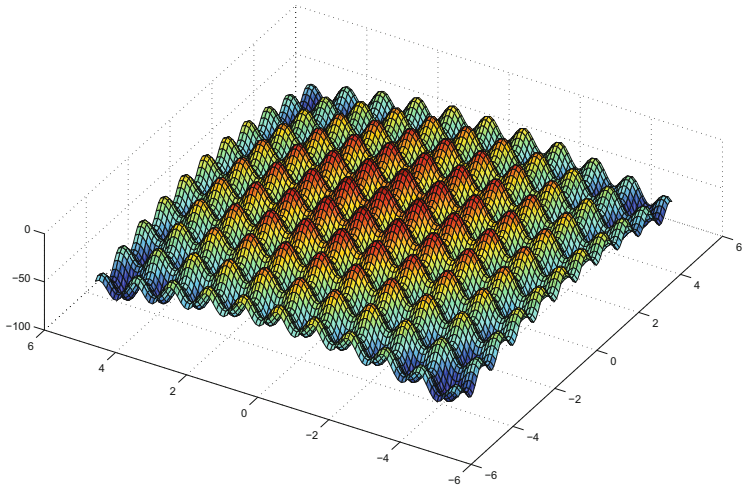


Fig. 1 Graph of $-r(x)$ for $n = 2$

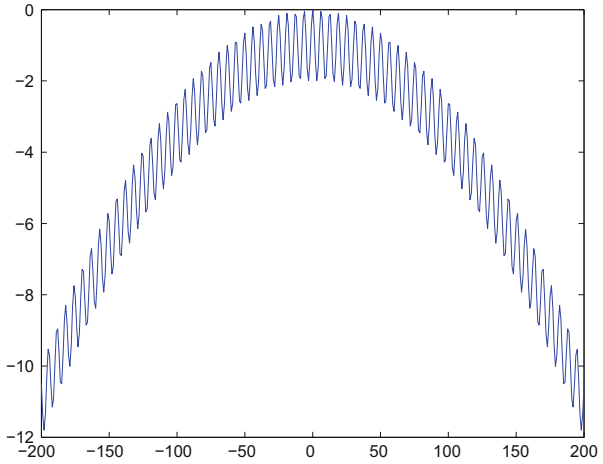


Fig. 2 Graph of $-g(x)$ for $n = 1$

where $x \in [-600, 600]^n$. This problem has one global solution defined at $x^* = (0, 0, \ldots, 0)$ with value $g(x^*) = 0$. The problem also has several local minima. This is illustrated by Figs. 2 and 3, for $n = 1$ and $n = 2$, respectively (the figures represents the graph of $-g(x)$).

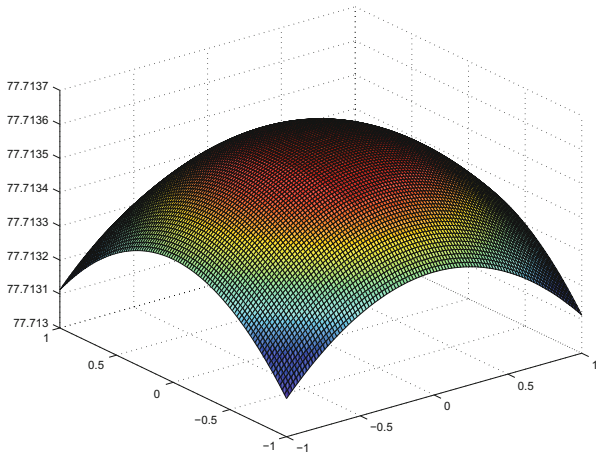


Fig. 3 Graph of $-g(x)$ for $n = 2$

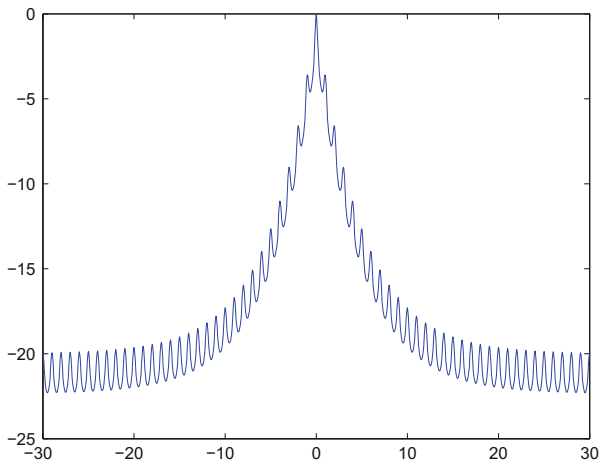


Fig. 4 Graph of $-a(x)$ for $n = 1$

4.2.4 Ackley Function

The Ackley optimization problem can be defined, for a fixed n, as

$$\min a(x) \equiv -20e^{-0.02\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$$

where $x \in [-30, 30]^n$. This problem has one global solution defined at $x^* = (0, 0, \ldots, 0)$ with value $a(x^*) = 0$. For $n = 1$ and $n = 2$, the problem has several minima, as can be seen in Figs. 4 and 5.

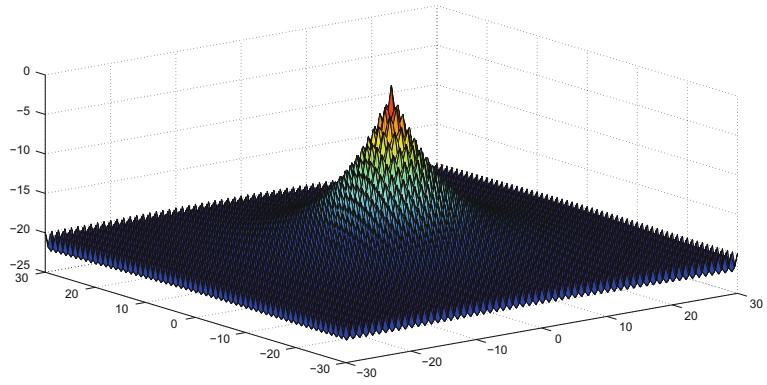


Fig. 5 Graph of $-a(x)$ for $n = 2$

4.3 Search Times and Speedups

The first set of experimental results presented and discussed are the optima search times (T), and the speedups (S) achieved by the various PSSA approaches. Search times were measured both for the PSSA approaches and also for the original SSA. In the later case, the values presented, denoted by T_{SSA}, are the times consumed in order to conduct the optima search sequentially in all subdomains (i.e., one subdomain after the other). Therefore, T_{SSA} provides a comparison baseline from which one can derive the performance speedups attained by the PSSA parallelization strategies.

The analysis of search times and speedups is separated in two different subsections, because, depending on the kind of domain decomposition, the overall number of subdomains searched, s, varies differently: with *homogeneous decomposition*, $s = s_{init}$ and so s is completely deterministic (as given by formulation (7), s_{init} depends only on the parameters g and n, initially fixed); with *heterogeneous decomposition*, s_{init} is just a lower bound for s and, due to search recursion, usually $s \gg s_{init}$.

4.3.1 Homogeneous Decomposition

Tables 2, 3, 4, and 5 show the search times (in seconds) and speedups with *homogeneous decomposition*, for the studied problems, with various decomposition granularities.

The tables reveal that the parallel approaches PSSA_HoS and PSSA_HoD ensure significantly lower search times in comparison to SSA_Ho, which denotes the original SSA applied, in sequence (serially) to the same set of homogeneous subdomains.

Table 2 Search times and speedups for Hartman, with *homogeneous decomposition*

g	s	T_{SSA_Ho}	T_{PSSA_HoS}	T_{PSSA_HoD}	S_{ideal}	S_{PSSA_HoS}	S_{PSSA_HoD}
1/1	1	1.24	1.17	1.17	1	1.06	1.06
1/2	8	12.22	2.99	3.01	8	4.09	4.06
1/4	64	93.01	8.69	7.79	15	10.70	11.94
1/8	512	697.32	49.93	47.79	15	13.97	14.59
1/16	4096	4820.83	345.70	324.39	15	13.95	14.86
1/32	32,768	35,772.35	2300.73	2247.45	15	15.55	15.92
Average		6899.50	451.54	438.60		9.88	10.41

Table 3 Search times and speedups for Rastrigin, with *homogeneous decomposition*

g	s	T_{SSA_Ho}	T_{PSSA_HoS}	T_{PSSA_HoD}	S_{ideal}	S_{PSSA_HoS}	S_{PSSA_HoD}
1/1	1	2.76	2.83	2.75	1	0.98	1.00
1/2	4	10.39	2.77	2.77	4	3.75	3.75
1/4	16	80.52	9.57	7.54	15	8.41	10.68
1/8	64	335.31	27.48	26.40	15	12.20	12.70
1/16	256	906.87	71.40	62.20	15	12.70	14.58
1/32	1024	1832.77	145.01	124.76	15	12.64	14.69
Average		528.10	43.18	37.74		8.45	9.57

Table 4 Search times and speedups for Griewank, with *homogeneous decomposition*

g	s	T_{SSA_Ho}	T_{PSSA_HoS}	T_{PSSA_HoD}	S_{ideal}	S_{PSSA_HoS}	S_{PSSA_HoD}
1/1	1	5.72	5.74	5.74	1	1.00	1.00
1/2	4	17.15	5.86	5.82	4	2.93	2.95
1/4	16	74.96	8.95	8.93	15	8.93	8.95
1/8	64	251.65	23.33	21.46	15	10.79	11.73
1/16	256	836.70	63.50	58.77	15	13.18	14.24
1/32	1024	2819.53	198.46	194.69	15	14.21	14.48
Average		668.45	50.97	49.24		8.50	8.89

It may also be concluded that PSSA_HoD is slightly faster than PSSA_HoS: on average, the ratio $T_{PSSA\_HoD}/T_{PSSA\_HoD}$ ranges from $438.60/451.54 = 97.1\,\%$ (for the Hartman function) down to $37.74/43.18 = 87.4\,\%$ (for the Rastrigin function).

The performance advantage of PSSA_HoD over PSSA_HoS also emerges when comparing their speedups on SSA_Ho. These speedups are given by $S_{PSSA\_HoD} = \frac{T_{SSA\_Ho}}{T_{PSSA\_HoD}}$ and $S_{PSSA\_HoS} = \frac{T_{SSA\_Ho}}{T_{PSSA\_HoS}}$, respectively. On average, S_{PSSA_HoD} is always above S_{PSSA_HoS}: the ratio $S_{PSSA\_HoD}/S_{PSSA\_HoD}$ ranges from $8.89/8.50 = 104.5\,\%$ (for the Griewank function) up to $9.57/8.45 = 113.2\,\%$ (for the Rastrigin function).

Search times and speedups are plotted in Figs. 6 to 7, and 8 to 9, respectively.

In the search time graphics, the differences between the times of the different parallel approaches are hardly visible (the lines mostly overlap), despite the use of vertical axis breaks. This is due to the considerable gap (of at least one order

Table 5 Search times and speedups for Ackley, with *homogeneous decomposition*

g	s	T_{SSA_Ho}	T_{PSSA_HoS}	T_{PSSA_HoD}	S_{ideal}	S_{PSSA_HoS}	S_{PSSA_HoD}
1/1	1	1.09	1.09	1.09	1	1.00	1.00
1/2	4	20.16	7.26	7.57	4	2.78	2.66
1/4	16	88.98	12.40	8.23	15	7.18	10.81
1/8	64	363.70	33.01	29.90	15	11.02	12.16
1/16	256	1372.94	97.83	98.70	15	14.03	13.91
1/32	1024	4975.80	345.35	333.57	15	14.41	14.92
Average		1137.11	82.82	79.84		8.40	9.24

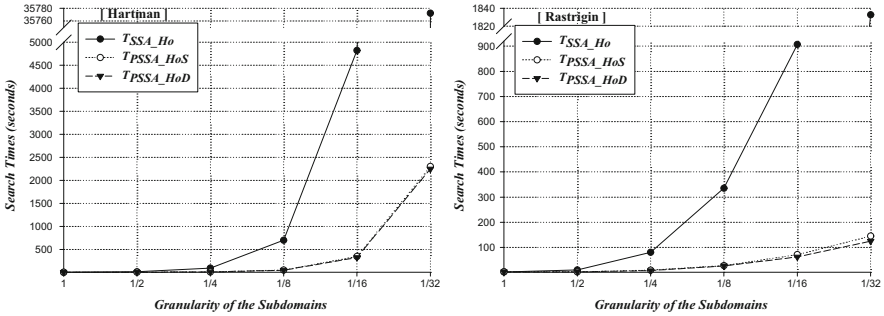


Fig. 6 Search times for Hartman and Rastrigin, with *homogeneous decomposition*

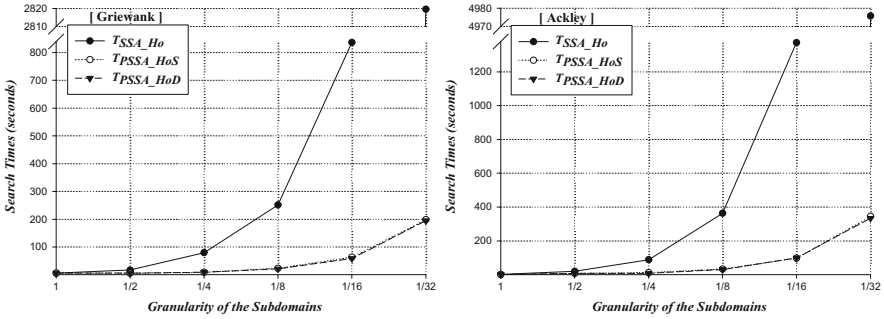


Fig. 7 Search times for Griewank and Ackley, with *homogeneous decomposition*

of magnitude) separating parallel from serial times. In this regard, the differences among the parallel approaches are only perceivable by inspecting the result tables.

The speedup graphics include an ideal speedup, represented by S_{ideal} (also present in the tables). The later metric basically establishes an upper bound for the achievable speedup when searching s subdomains in parallel using one or more of the 15 spawned *slaves*. Thus, the values of S_{ideal} match the number of *slaves* that, in the testbed cluster, are actively engaged in optima search, for a certain number s of subdomains: if $s = 1$, then $S_{ideal} = 1$, once only 1 *slave* will be necessary (the

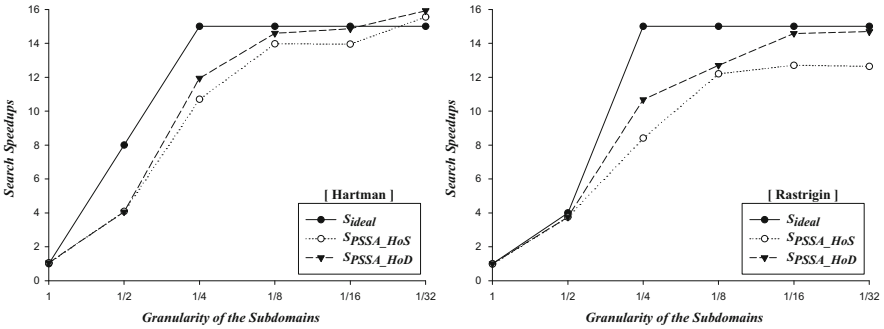


Fig. 8 Speedups for Hartman and Rastrigin, with *homogeneous decomposition*

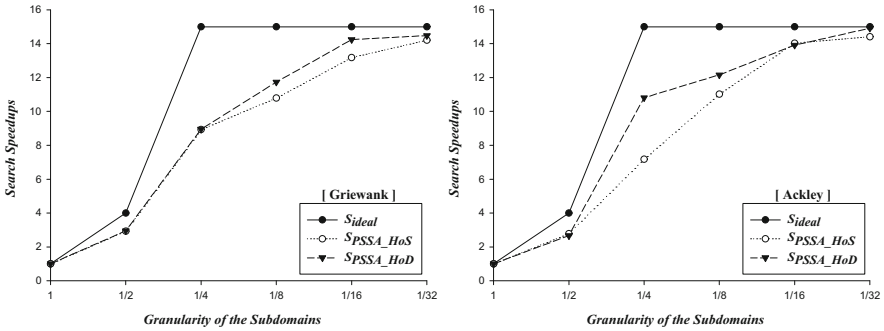


Fig. 9 Speedups for Griewank and Ackley, with *homogeneous decomposition*

other 14 will be idle); if $s = 4$, then $S_{ideal} = 4$, once only 4 *slaves* will be needed; if $s = 8$, then $S_{ideal} = 8$, once only 8 *slaves* will be busy; finally, whenever $s \geq 15$, all *slaves* will be necessary, and the maximum expected speedup will be $S_{ideal} = 15$.

The upper bound provided by S_{ideal} for the parallel search speedup is not, however, a hard limit: sometimes, it may be surpassed. Such happens with the Hartman problem, a function of $n = 3$ dimensions, that generates a high number of very small subdomains when the decomposition granularity (g) becomes very small; in these situations, subdomains are so small that they are quickly dismissed by SSA; the cumulative effect of this fast searches happening in parallel leads to "above linear scalability", translating in higher than expected speedups, a situation know as "super-scalability". In the experiments with the Hartman function, this phenomena was observed with $s = 32{,}768$ subdomains (or, equivalently, when $g = 1/32$).

4.3.2 Heterogeneous Decomposition

Tables 6, 7, 8, and 9 show the search times and respective speedups for two approaches based on *heterogeneous decomposition*: the parallel approach

Table 6 Search times and speedups for Hartman, with *heterogeneous decomposition*

g	s_{init}	T_{SSA_He} $(h = 2)$	T_{SSA_He} $(h = 0)$	T_{PSSA_HeD} $(h = 2)$	T_{PSSA_HeD} $(h = 0)$	S^{min}_{ideal}	S_{PSSA_HeD} $(h = 2)$	S_{PSSA_HeD} $(h = 0)$
1/1	1	8.61	8.77	2.94	2.94	1	2.93	2.98
1/2	8	27.27	28.15	5.03	5.27	8	5.17	5.60
1/4	64	110.11	110.31	7.69	7.81	15	14.32	14.12
1/8	512	681.70	689.50	47.19	47.65	15	14.45	14.47
1/16	4096	4822.37	5101.44	324.23	326.13	15	14.87	15.64
1/32	32,768	33,553.93	33,752.12	2244.39	2248.19	15	14.95	15.01
Average		6613.72	6615.05	438.58	439.67		11.16	11.26

Table 7 Search times and speedups for Rastrigin, with *heterogeneous decomposition*

g	s_{init}	T_{SSA_He} $(h = 2)$	T_{SSA_He} $(h = 0)$	T_{PSSA_HeD} $(h = 2)$	T_{PSSA_HeD} $(h = 0)$	S^{min}_{ideal}	S_{PSSA_HeD} $(h = 2)$	S_{PSSA_HeD} $(h = 0)$
1/1	1	59.75	78.53	11.93	12.04	1	5.01	6.52
1/2	4	82.63	82.77	9.29	9.27	4	8.89	8.93
1/4	16	415.10	406.14	27.95	28.49	15	14.85	14.26
1/8	64	1645.56	1652.75	113.62	115.91	15	14.48	14.26
1/16	256	3207.77	3211.68	219.93	221.59	15	14.59	14.49
1/32	1024	4154.30	4150.53	280.55	282.76	15	14.81	14.68
Average		1594.19	1597.07	110.55	111.68		12.11	12.19

Table 8 Search times and speedups for Griewank, with *heterogeneous decomposition*

g	s_{init}	T_{SSA_He} $(h = 2)$	T_{SSA_He} $(h = 0)$	T_{PSSA_HeD} $(h = 2)$	T_{PSSA_HeD} $(h = 0)$	S^{min}_{ideal}	S_{PSSA_HeD} $(h = 2)$	S_{PSSA_HeD} $(h = 0)$
1/1	1	385.29	1025.34	37.91	112.66	1	10.16	9.10
1/2	4	375.71	1719.51	32.72	108.45	4	11.48	15.86
1/4	16	526.15	1866.75	41.95	131.89	15	12.54	14.15
1/8	64	1231.37	2942.12	88.03	175.29	15	13.99	16.78
1/16	256	3484.63	6022.39	234.04	425.85	15	14.89	14.14
1/32	1024	10,012.68	16,120.09	685.68	1077.62	15	14.60	14.96
Average		2669.31	4949.37	186.72	338.63		12.94	14.17

PSSA_HeD (presented in Sect. 3.4), and its serial counterpart, named SSA_He, that applies SSA, in sequence, to the heterogeneous subdomains. This serial variant is necessary as a comparison baseline to evaluate PSSA_HeD, as it would be inappropriate to use SSA_Ho, based on homogeneous subdomains, for the same purpose.

Moreover, both SSA_He and PSSA_HeD were studied with two different values of the height parameter that controls the depths of the recursive search: $h = 2$ and $h = 0$. These specific values for h were chosen in order to discover if conducting an exhaustive search (when $h = 0$) really payed off in comparison with a very

Table 9 Search times and speedups for Ackley, with *heterogeneous decomposition*

g	s_{init}	T_{SSA_He} $(h = 2)$	T_{SSA_He} $(h = 0)$	T_{PSSA_HeD} $(h = 2)$	T_{PSSA_HeD} $(h = 0)$	S^{min}_{ideal}	S_{PSSA_HeD} $(h = 2)$	S_{PSSA_HeD} $(h = 0)$
1/1	1	19.46	19.02	6.92	6.95	1	2.81	2.74
1/2	4	160.26	160.06	20.31	18.15	4	7.89	8.82
1/4	16	1133.92	2545.51	89.29	165.18	15	12.70	15.41
1/8	64	4392.66	5449.28	282.64	345.86	15	15.54	15.76
1/16	256	12,799.00	12,721.95	811.51	853.26	15	15.77	14.91
1/32	1024	30,152.16	30,329.25	2011.74	2043.42	15	14.99	14.84
Average		8109.58	8537.51	537.07	572.14		11.62	12.08

superficial one (one level bellow the starting grid of subdomains, as dictated by $h = 2$).

When comparing the search times $T_{PSSA\_HeD(h=2)}$ with $T_{SSA\_He(h=2)}$, and $T_{PSSA\_HeD(h=0)}$ with $T_{SSA\_He(h=0)}$, the parallel approaches are again much faster than their serial baselines. And, for all the four functions studied, the fastest approach based on *heterogeneous decomposition* is, on average, *PSSA_HeD*$(h = 2)$, which is really expected: it is a parallel approach and its search depth is the smallest possible.

The relative performance advantage of the parallel approaches over the serial ones is given by the speedups $S_{PSSA\_HeD(h=2)} = T_{SSA\_He(h=2)}/T_{PSSA\_HeD(h=2)}$ and $S_{PSSA\_HeD(h=0)} = T_{SSA\_He(h=0)}/T_{PSSA\_HeD(h=2)}$. In this regard, $S_{PSSA\_HeD(h=0)} > S_{PSSA\_HeD(h=2)}$, on average. But this doesn't mean that *PSSA_HeD*$(h = 0)$ is more attractive than *PSSA_HeD*$(h = 2)$, performance-wise. In fact, $T_{PSSA\_HeD(h=0)} > T_{PSSA\_HeD(h=2)}$, on average. This misalignment between speedups and search times is easily explainable: *PSSA_HeD*$(h = 0)$ will typically generate and process (much) more subdomains than *PSSA_HeD*$(h = 2)$, thus taking more time; on the other hand, as more subdomains are generated, these tend to be more fine-grained, which leads to a better workload distribution among slaves, thus translating in *PSSA_HeD*$(h = 0)$ having better speedups over its serial baseline *SSA_He*$(h = 0)$, than *PSSA_HeD*$(h = 2)$ over the serial baseline *SSA_He*$(h = 2)$. The fact that *PSSA_HeD*$(h = 0)$ and *PSSA_HeD*$(h = 2)$ have different baselines also prevents a direct comparison between their speedups (this is in contrast with *PSSA_HoS* and *PSSA_HoD*, whose speedups were comparable because they shared the same baseline, *SSA_Ho*). This also means that additional metrics, of a different nature, are needed, in order to compare the various parallel approaches so far developed (see Sect. 4.6).

Figures 10 to 11, and 12 to 13, plot the search times and speedups, respectively.

Again, the lines in the graphics split into two sets of different orders of magnitude, one for serial search times, another for parallel search times. Each set comprises two lines, one for the search depth $h = 0$, another for $h = 2$. With the exception of the Griewank function, the lines for the same h value mostly overlap, i.e., the variation of h has little or no influence on the search time (as may also be confirmed in the tables, namely by comparing average search times). The

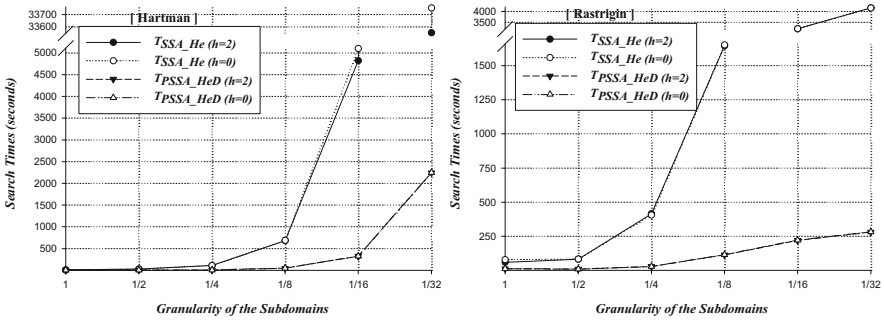


Fig. 10 Search times for Hartman and Rastrigin, with *heterogeneous decomposition*

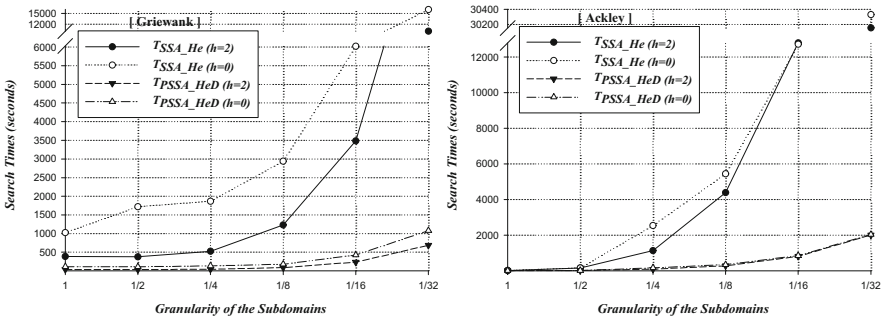


Fig. 11 Search times for Griewank and Ackley, with *heterogeneous decomposition*

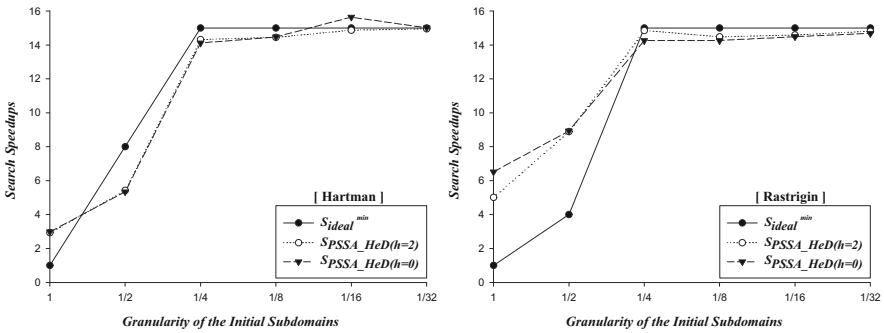


Fig. 12 Speedups for Hartman and Rastrigin, with *heterogeneous decomposition*

explanation for this may be found in the experimental data further discussed in Sect. 4.4: the maximum effective search depth reached for the functions Hartman and Rastrigin, when setting $h = 0$, is in fact $h_{max} = 2$ (on average) and, for function Ackley, is $h_{max} = 3$ (on average); therefore, the overall number of subdomains searched when initially setting $h = 0$ is similar (or not very far, for the Ackley

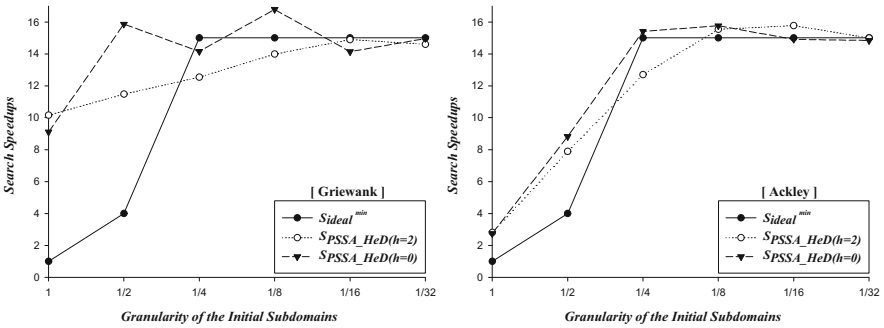


Fig. 13 Speedups for Griewank and Ackley, with *heterogeneous decomposition*

function) to the overall number searched when setting $h = 2$, thus translating in similar search times.

The speedup graphics include S_{ideal}^{min} (also present in the tables), a lower bound for the ideal speedup. Indeed, with *heterogeneous decomposition* it is not possible to know, in anticipation, the overall number of subdomains to be analyzed, once new subdomains are dynamically generated, making impossible to predict the correct ideal speedup; only the number of initial subdomains, s_{init}, is deterministic and, together with the overall number of spawned slaves, may be used to derive S_{ideal}^{min}.

4.4 Number of Subdomains Searched and Maximum Search Depth

This section discusses results that are solely relevant when pursuing *heterogeneous decomposition*. These results include the (overall) number of subdomains searched, s, and the maximum search depth achieved, h^{max}. With *homogeneous decomposition*, such metrics are deterministic and initially fixed: as previously stated, $s = s_{init}$; also, $h^{max} = 1$, once no further subdomain split takes place bellow the original grid of homogeneous subdomains. However, with *heterogeneous decomposition*, the final value of s is unpredictable; also, when $h = 0$, the value of h^{max} is also unpredictable (note that h^{max} is deterministic only when $h \neq 0$, in which case $h^{max} = h$).

Tables 10, 11, 12, and 13 show s and h^{max} for the *heterogeneous decomposition* approaches.

The tables show that the final number of subdomains searched in all approaches ($s_{SSA\_He(h=0)}$, $s_{SSA\_He(h=0)}$, $s_{PSSA\_HeD(h=0)}$ and $s_{PSSA\_HeD(h=2)}$) is considerably larger than the initial number, s_{init} (though their ratios vary depending on the function).

Moreover, it may also observed that when s_{init} increases, a considerable amount of additional subdomains is still being generated, which can only happen because there are enough optima found to generate those new additional subdomains. In

Table 10 Number of subdomains searched and maximum search depth for Hartman, with *heterogeneous decomposition*

g	s_{init}	s_{SSA_He} $(h=2)$	s_{SSA_He} $(h=0)$	s_{PSSA_HeD} $(h=2)$	s_{PSSA_HeD} $(h=0)$	$h^{max}_{SSA\_He}$ $(h=0)$	$h^{max}_{PSSA\_HeD}$ $(h=0)$
1/1	1	9	9	9	9	2	2
1/2	4	24	24	24	24	2	2
1/4	16	88	88	88	88	2	2
1/8	64	536	536	536	536	2	2
1/16	256	4120	4120	4120	4120	2	2
1/32	1024	32,792	32,792	32,792	32,792	2	2
Average		**6261.5**	**6261.5**	**6261.5**	**6261.5**	2	2

Table 11 Number of subdomains searched and maximum search depth for Rastrigin, with *heterogeneous decomposition*

g	s_{init}	s_{SSA_He} $(h=2)$	s_{SSA_He} $(h=0)$	s_{PSSA_HeD} $(h=2)$	s_{PSSA_HeD} $(h=0)$	$h^{max}_{SSA\_He}$ $(h=0)$	$h^{max}_{PSSA\_HeD}$ $(h=0)$
1/1	1	13	13	13	13	2	2
1/2	4	18	18	14	14	2	2
1/4	16	76	76	72	72	2	2
1/8	64	310	310	310	316	2	2
1/16	256	697	697	697	697	2	2
1/32	1024	1465	1465	1465	1465	2	2
Average		**429.83**	**429.83**	428.50	**429.50**	2.00	2.00

Table 12 Number of subdomains searched and maximum search depth for Griewank, with *heterogeneous decomposition*

g	s_{init}	s_{SSA_He} $(h=2)$	s_{SSA_He} $(h=0)$	s_{PSSA_HeD} $(h=2)$	s_{PSSA_HeD} $(h=0)$	$h^{max}_{SSA\_He}$ $(h=0)$	$h^{max}_{PSSA\_HeD}$ $(h=0)$
1/1	1	78	205	78	271	15	18
1/2	4	72	337	76	220	21	15
1/4	16	105	365	116	333	15	15
1/8	64	300	659	288	526	24	12
1/16	256	944	1504	955	1533	13	15
1/32	1024	3237	4566	3249	4529	10	9
Average		789.33	**1272.67**	793.67	**1235.33**	16.33	14.00

other words, this means that decreasing the granularity g of the starting grid of subdomains leads to more optima being found (confirmed by results of Sect. 4.5).

The variation of the search depth limit ($h = 0$ vs $h = 2$) has a different effect on the number of subdomains searched, depending on the function tested. Thus, for the Hartman function, the number of subdomains searched is not affected whatsoever: those numbers match, for the same g and h, across all heterogeneous search approaches, whether serial or parallel; the explanation resides on the fact

Table 13 Number of subdomains searched and maximum search depth for Ackley, with *heterogeneous decomposition*

g	s_{init}	s_{SSA_He} $(h = 2)$	s_{SSA_He} $(h = 0)$	s_{PSSA_HeD} $(h = 2)$	s_{PSSA_HeD} $(h = 0)$	$h^{max}_{SSA\_He}$ $(h = 0)$	$h^{max}_{PSSA\_HeD}$ $(h = 0)$
1/1	1	5	5	5	5	2	2
1/2	4	28	28	30	36	2	3
1/4	16	184	417	202	390	7	7
1/8	64	745	913	714	879	5	4
1/16	256	2169	2275	2151	2276	4	4
1/32	1024	5759	5793	5725	5786	3	3
Average		1481.67	**1571.83**	1471.17	**1562**	3.83	3.83

that, for Hartman, the maximum search depth achieved is still 2 when allowed to be unlimited (see columns $h^{max}_{SSA\_He}(h = 0)$ and $h^{max}_{PSSA\_HeD}(h = 0)$ of Table 10). For the Rastrigin function (Table 11), the variation of h also has almost no influence on the number of subdomains searched and, again, the maximum search depth achieved is also 2. In turn, for the Ackley function (Table 13), setting $h = 0$ typically leads to more subdomains searched than with $h = 2$, and now the maximum search depths ranges from 2 up to 7. Finally, for the Griewank function, the growing in the number of subdomains searched when $h = 0$, in comparison to when $h = 2$, is much more evident, translating into maximum search depths that range from 9 to 24.

A result that deserves further investigation is the fact that, sometimes, a serial approach generates and searches slightly more subdomains than its parallel version (although, of course, at the cost of much more search time). For instance, with the Ackley function, $s_{SSA\_He}(h = 2) = 1481.67 > s_{PSSA\_HeD}(h = 2) = 1471.17$ (on average), as well as $s_{SSA\_He}(h = 0) = 1571.83 > s_{PSSA\_HeD}(h = 0) = 1562$ (on average).

4.5 Number of Optima Found

So far, the discussion of the experimental data has been made from a computational/algorithmic perspective, centered on performance results (search times and speedups) and other metrics (number of subdomains searched and maximum search depth). The focus now turns to the number of optima found, denoted by n^*, a metric that conveys the efficiency of the search process from a numerical perspective.

Tables 14, 15, 16, and 17 show the number of optima found for each one of the functions studied, based on the search conducted by all serial and parallel approaches.

In the previous tables it may be observed that the number of optima found (n^*) grows when the granularity (g) decreases, for any problem and search approach (an auxiliary metric is provided, based on the sum of all optima found will all

Table 14 Number of optima found for Hartman, with all evaluated approaches

g	$n^*_{SSA\_Ho}$	$n^*_{PSSA\_HoS}$	$n^*_{PSSA\_HoD}$	$n^*_{SSA\_He}$ $(h = 2)$	$n^*_{SSA\_He}$ $(h = 0)$	$n^*_{PSSA\_HeD}$ $(h = 2)$	$n^*_{PSSA\_HeD}$ $(h = 0)$
1/1	1	1	1	1	1	1	1
1/2	2	2	2	2	2	2	2
1/4	3	3	3	3	3	3	3
1/8	3	3	3	3	3	3	3
1/16	4	4	4	4	4	4	4
1/32	38	38	38	38	38	38	38
Sum	**51**	**51**	**51**	**51**	**51**	**51**	**51**

Table 15 Number of optima found for Rastrigin, with all evaluated approaches

g	$n^*_{SSA\_Ho}$	$n^*_{PSSA\_HoS}$	$n^*_{PSSA\_HoD}$	$n^*_{SSA\_He}$ $(h = 2)$	$n^*_{SSA\_He}$ $(h = 0)$	$n^*_{PSSA\_HeD}$ $(h = 2)$	$n^*_{PSSA\_HeD}$ $(h = 0)$
1/1	3	3	3	4	4	3	3
1/2	7	5	5	7	7	5	5
1/4	22	22	22	22	22	22	22
1/8	76	74	75	76	76	76	78
1/16	121	121	121	121	121	121	121
1/32	121	121	121	121	121	121	121
Sum	**350**	346	347	**351**	**351**	348	**350**

Table 16 Number of optima found for Griewank, with all evaluated approaches

g	$n^*_{SSA\_Ho}$	$n^*_{PSSA\_HoS}$	$n^*_{PSSA\_HoD}$	$n^*_{SSA\_He}$ $(h = 2)$	$n^*_{SSA\_He}$ $(h = 0)$	$n^*_{PSSA\_HeD}$ $(h = 2)$	$n^*_{PSSA\_HeD}$ $(h = 0)$
1/1	18	18	18	24	51	24	68
1/2	22	23	23	34	93	35	63
1/4	31	34	34	51	102	50	94
1/8	72	73	69	105	168	106	133
1/16	182	184	187	242	331	271	342
1/32	572	559	576	744	921	792	919
Sum	897	891	907	1200	**1666**	1278	**1619**

granularities). It may also be observed that: (i) *PSSA_HeD*$(h = 0)$ is the parallel approach that returns more optima; (ii) *SSA_He*$(h = 0)$ is a serial approach that usually finds even more optima than the best parallel approach; (iii) the best serial and parallel approaches are both based on *heterogeneous decomposition* with infinite search depth ($h = 0$). By comparing these tables with Tables 10, 11, 12, and 13 it may be concluded that the approaches that generate more subdomains are also the ones that find more optima, as expected. The Hartman function still exhibits a special behavior: Table 10 showed the overall number of subdomains searched to be

Table 17 Number of optima found for Ackley, with all evaluated approaches

g	$n^*_{SSA\_Ho}$	$n^*_{PSSA\_HoS}$	$n^*_{PSSA\_HoD}$	$n^*_{SSA\_He}$ $(h = 2)$	$n^*_{SSA\_He}$ $(h = 0)$	$n^*_{PSSA\_HeD}$ $(h = 2)$	$n^*_{PSSA\_HeD}$ $(h = 0)$
1/1	1	1	1	1	1	1	1
1/2	8	9	9	12	12	15	16
1/4	60	55	57	99	154	99	142
1/8	242	232	233	296	309	294	300
1/16	693	682	676	734	741	739	737
1/32	1432	1420	1425	1454	1454	1427	1443
Sum	2436	2399	2401	2596	**2671**	2575	**2639**

the same, on average, for all approaches, and the same happens with the number of optima found in Table 14.

4.6 Selecting a Specific Approach

The search times (T) presented in Sect. 4.3, and the number of optima found (n^*) presented in the previous section, may be used, separately, in order to choose the most effective approach when considering a single of those factors. However, in some situations a compromise may be desirable, which demands a combined metric.

In this context, "attractiveness" is introduced, as a linear metric given by

$$A_a = \alpha \times (1 - \frac{T_a}{T_{max}}) + \beta \times \frac{n^*_a}{n^*_{max}}, \tag{8}$$

where

- $A_a \in [0, 1]$ is the attractiveness of an approach $a \in \{SSA\_Ho, PSSA\_HoS, PSSA\_HoD, SSA\_He(h = 2), SSA\_He(h = 0), PSSA\_HeD(h = 2), PSSA\_HeD(h = 0)\}$;
- α and β are complementary weights, such that $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$;
- T_a is the search time of a specific approach a;
- T_{max} is the maximum of all the search times(T_a) of all approaches;
- n^*_a is the number of optima found by a specific approach a;
- n^*_{max} is the maximum of all the numbers of optima (n^*_a) of all approaches.

A_a is basically a weighted average of contributions from the search times domain, and the number of optima domain. This average may be tunned accordingly with the relevance chosen for each factor. When $\alpha = 1$, the metric will produce a relative ranking that matches the absolute ranking based exclusively on the search times presented in Sect. 4.3. When $\alpha = 0$, the relative ranking generated will match the absolute ranking based on the number of optima found presented in Sect. 4.5.

Table 18 Attractiveness for Hartman, with $\alpha = \beta = 0.5$

g	A_{SSA_Ho}	A_{PSSA_HoS}	A_{PSSA_HoD}	A_{SSA_He} $(h = 2)$	A_{SSA_He} $(h = 0)$	A_{PSSA_HeD} $(h = 2)$	A_{PSSA_HeD} $(h = 0)$
1/1	0.51	0.51	0.51	0.51	0.51	0.51	0.51
1/2	0.53	0.53	0.53	0.53	0.53	0.53	0.53
1/4	0.54	0.54	0.54	0.54	0.54	0.54	0.54
1/8	0.53	0.54	0.54	0.53	0.53	0.54	0.54
1/16	0.49	0.55	0.55	0.49	0.48	0.55	0.55
1/32	0.50	0.97	0.97	0.53	0.53	0.97	0.97
Average	0.52	**0.61**	**0.61**	0.52	0.52	**0.61**	**0.61**

Table 19 Attractiveness for Rastrigin, with $\alpha = \beta = 0.5$

g	A_{SSA_Ho}	A_{PSSA_HoS}	A_{PSSA_HoD}	A_{SSA_He} $(h = 2)$	A_{SSA_He} $(h = 0)$	A_{PSSA_HeD} $(h = 2)$	A_{PSSA_HeD} $(h = 0)$
1/1	0.51	0.51	0.51	0.51	0.51	0.51	0.51
1/2	0.53	0.52	0.52	0.52	0.52	0.52	0.52
1/4	0.58	0.59	0.59	0.54	0.54	0.59	0.59
1/8	0.77	0.80	0.81	0.62	0.62	0.80	0.81
1/16	0.89	0.99	0.99	0.61	0.61	0.97	0.97
1/32	0.78	0.98	0.98	0.50	0.50	0.97	0.97
Average	0.68	**0.73**	**0.73**	0.55	0.55	**0.73**	**0.73**

Table 20 Attractiveness for Griewank, with $\alpha = \beta = 0.5$

g	A_{SSA_Ho}	A_{PSSA_HoS}	A_{PSSA_HoD}	A_{SSA_He} $(h = 2)$	A_{SSA_He} $(h = 0)$	A_{PSSA_HeD} $(h = 2)$	A_{PSSA_HeD} $(h = 0)$
1/1	0.51	0.51	0.51	0.50	0.50	0.51	0.53
1/2	0.51	0.51	0.51	0.51	0.50	0.52	0.53
1/4	0.51	0.52	0.52	0.51	0.50	0.53	0.55
1/8	0.53	0.54	0.54	0.52	0.50	0.55	0.57
1/16	0.57	0.60	0.60	0.52	0.49	0.64	0.67
1/32	0.72	0.80	0.81	0.59	0.50	0.91	0.97
Average	0.56	0.58	0.58	0.53	0.50	0.61	**0.64**

Thus, the attractiveness metric becomes useful when the aim is to achieve a certain balance between the search time and the number of optima found. In order to illustrate the usefulness of the metric, Tables 18, 19, 20, and 21 provide results from its application when $\alpha = \beta = 0.5$, that is, when search time and number of optima have the same importance. The respective graphics are also presented (Figs. 14 and 15).

For this specific values of α and β, the following conclusions may be drawn: the *PSSA_HeD*$(h = 0)$ approach is, on average, the approach that consistently provides the highest levels of attractiveness independently of the optimization problem;

Table 21 Attractiveness for Ackley, with $\alpha = \beta = 0.5$

g	A_{SSA_Ho}	A_{PSSA_HoS}	A_{PSSA_HoD}	A_{SSA_He} $(h = 2)$	A_{SSA_He} $(h = 0)$	A_{PSSA_HeD} $(h = 2)$	A_{PSSA_HeD} $(h = 0)$
1/1	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1/2	0.50	0.50	0.50	0.50	0.50	0.50	0.51
1/4	0.52	0.52	0.52	0.52	0.51	0.53	0.55
1/8	0.58	0.58	0.58	0.53	0.52	0.60	0.60
1/16	0.72	0.73	0.73	0.54	0.55	0.74	0.74
1/32	0.91	0.98	0.98	0.50	0.50	0.96	0.96
Average	0.62	**0.64**	**0.64**	0.52	0.51	**0.64**	**0.64**

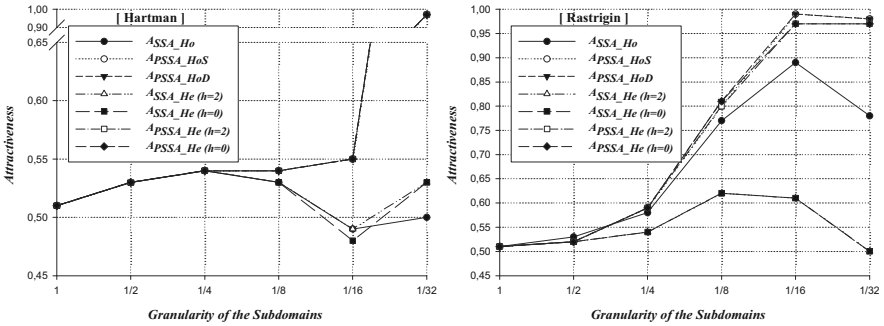


Fig. 14 Attractiveness for Hartman and Rastrigin, with $\alpha = \beta = 0.5$

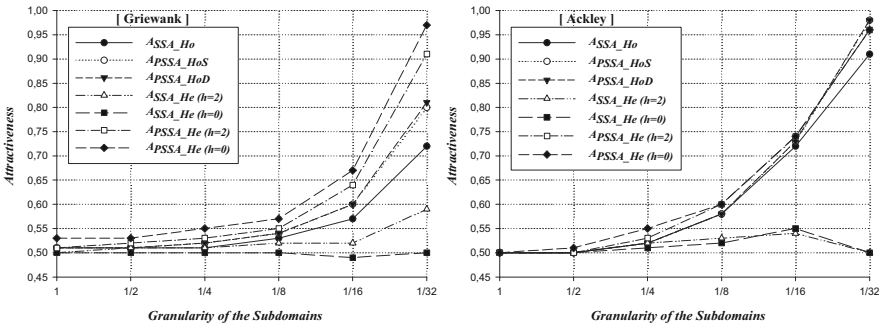


Fig. 15 Attractiveness for Griewank and Ackley, with $\alpha = \beta = 0.5$

nevertheless, for some specific problems (Hartman, Rastrigin, and Griewank), the attractiveness ensured by approaches *PSSA_HoS*, *PSSA_HoD* and *PSSA_HeD*$(h = 2)$ matches, on average, the one provided by *PSSA_HeD*$(h = 0)$; this happens because, for those functions, there's a compensation effect between the four approaches: for each approach that ensures low search times and low number of optima found, there will be other(s) with high search times and high number of optima found.

When working with a specific granularity (g), one should use the respective line in the tables as a guide to properly choose the most attractive approach, instead of recurring to the average values; in pursuing this, one may verify that, sometimes (for Rastrigin with $g = 1/16$ and $g = 1/32$, or Ackley with $g = 1/32$) *PSSA_HoS* and *PSSA_HoD* are more attractive than *PSSA_HeD*($h = 2$) and *PSSA_HeD*($h = 0$).

5 Conclusions and Future Work

In this work, PSSA was thoroughly described, as a set of parallel computing based approaches, aimed at improving the number of local solutions of multilocal optimization problems found by the SSA stochastic algorithm. All PSSA approaches build on the Data Decomposition parallelization paradigm, applied to the feasible region of the optimization problem. However, the various approaches presented (PSSA_HoS, PSSA_HoD and PSSA_HeD) exhibit increased algorithmic complexity, covering a wide range of parallelization strategies that combine different domain decomposition techniques (homogeneous vs heterogeneous decomposition) with different work allocation techniques (static vs dynamic distribution).

The experimental examination of the developed approaches proves their scalability and shows that the most sophisticated approach PSSA_HeD, with heterogeneous decomposition,and dynamic distribution outperforms the others PSSA_HoS and PSSA_HoD, both with homogeneous decomposition, in numerical efficiency (number of optima found), although at the cost of higher search times.

In order to strike the right balance between numerical efficiency and search time, a linear metric was also introduced to assist the selection of the best PSSA approach.

Directions for future work include the refinement of PSSA in order to further improve its scalability, the investigation of possible hybrid/heterogeneous parallel computing methods to improve performance, and the exploration of PSSA as a software component to solve constrained multilocal optimization problems.

References

1. Chelouah, R., Siarry, P.: A continuous genetic algorithm designed for the global optimization of multimodal functions. J. Heuristics **6**, 191–213 (2000)
2. Eriksson, P., Arora, J.: A comparison of global optimization algorithms applied to a ride comfort optimization problem. Struct. Multidiscip. Optim. **24**, 157–167 (2002)
3. Floudas, C.: Recent advances in global optimization for process synthesis, design and control: enclosure of all solutions. Comput. Chem. Eng. vol. 23, S963–S973 (1999)
4. Guibas, L.J., Sedgewick, R.: A dichromatic framework for balanced trees. In: Proceedings of the 19th Annual Symposium on Foundations of Computer Science, Ann Arbor, pp. 8–21 (1978)
5. Hedar, A.R.: Global Optimization Test Problems (2015). http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm
6. High-Performance Portable MPI (2015) – http://www.mpich.org/
7. Ingber, L.: Very fast simulated re-annealing. Math. Comput. Model. **12**, 967–973 (1989)

8. Kernighan, B.W., Ritchie, D.M.: The C Programming Language, 2nd edn. Prentice Hall, Englewood Cliffs (1988). ISBN 0-13-110362-8
9. Kiseleva, E., Stepanchuk, T.: On the efficiency of a global non-differentiable optimization algorithm based on the method of optimal set partitioning. J. Glob. Optim. **25**, 209–235 (2003)
10. León, T., Sanmatías, S., Vercher, H.: A multi-local optimization algorithm. Top **6**(N. 1), 1–18 (1998)
11. Message Passing Interface Forum (2015) – http://www.mpi-forum.org/
12. Parsopoulos, K., Plagianakos, V., Magoulas, G., Vrahatis, M.: Objective function stretching to alleviate convergence to local minima. Nonlinear Anal. **47**, 3419–3424 (2001)
13. Parsopoulos, K., Vrahatis, M.: Recent approaches to global optimization problems through particle swarm optimization. Nat. Comput. **1**, 235–306 (2002)
14. Pereira, A.I., Fernandes, E.M.G.P.: A reduction method for semi-infinite programming by means of a global stochastic approach. Optimization **58**, 713–726 (2009)
15. Pereira, A.I., Ferreira, O., Pinho, S.P., Fernandes, E.M.G.P.: Multilocal programming and applications. In: Zelinka, I., Snasel, V., Abraham, A. (eds.) Handbook of Optimization. Intelligent Systems Series, pp. 157–186. Springer, Berlin/New York (2013)
16. Pereira, A.I., Fernandes, E.M.G.P.: Constrained Multi-global optimization using a penalty stretched simulated annealing framework. In: Numerical Analysis and Applied Mathematics. AIP Conference Proceedings, Crete, vol. 1168, pp. 1354–1357 (2009)
17. Pereira, A.I., Fernandes, E.M.G.P.: Comparative study of penalty simulated annealing methods for multiglobal programming. In: 2nd International Conference on Engineering Optimization, Lisbon (2010)
18. Price, C.: Non-linear Semi-infinite Programming. University of Canterbury (1992)
19. Rauber, T., Runger, G.: Parallel Programming for Multicore and Cluster Systems. Springer (2010). ISBN 978-3-642-04817-3
20. Ribeiro, T., Rufino, J., Pereira, A.I.: PSSA: parallel stretched simulated annealing. In: Proceedings of the 2011' International Conference on Numerical Analysis and Applied Mathematics, Halkidiki, pp. 783–786 (2011)
21. Salhi, S., Queen, N.: A hybrid algorithm for identifying global and local minima when optimizing functions with many minima. Eur. J. Oper. Res. **155**, 51–67 (2004)
22. Snir, M., Otto, S.W., Huss-Lederman, S., Walker, D.W.: MPI-The Complete Reference (Volume 1). MIT, Cambridge (1988). ISBN 0-262-69215-5
23. The GNU C Library (2015) – http://www.gnu.org/software/libc/manual/
24. TOP500 Supercomputer Sites (2015) – http://www.top500.org
25. Tsoulos, I., Lagaris, I.: Gradient-controlled, typical-distance clustering for global optimization (2004). www.optimization.org
26. Tu, W., Mayne, R.: Studies of multi-start clustering for global optimization. Int. J. Numer. Methods Eng. **53**, 2239–2252 (2002)

Efficiency and Productivity Assessment of Wind Farms

Clara Bento Vaz and Ângela Paula Ferreira

Abstract This study develops a framework to provide insights regarding the performance of the farms of an energy player in the Portuguese wind sector. The focus of the wind farm performance assessment is on the operating stage which corresponds to the electrical energy generation process, during 2010 and 2011. In a first stage, Data Envelopment Analysis is used to measure the efficiency of wind farms in generating electrical energy from the resources available and non-discretionary variables. This analysis enables the identification of the best practices of the efficient farms which can be emulated by inefficient ones. In a second stage, changes in wind farms productivity are investigated using Malmquist index. Bootstrap procedures are applied to obtain statistical inference on the efficiency estimates. We conclude that almost all farms decreased overall productivity levels, mainly due to the decline in the productivity levels of the frontier, which is in accordance with the decrease in wind availability observed in 2011.

1 Introduction

Data Envelopment Analysis (DEA) has been accepted as an important approach for performance assessment and benchmarking in several sectors [21]. Studies using DEA to assess the performance of wind farms are still scarce [12].

C.B. Vaz (✉)
Polytechnic Institute of Bragança, Campus de Santa Apolónia, Apartado 1134, 5301-857, Bragança, Portugal

CGEI / INESC TEC - INESC Technology and Science, Porto, Portugal
e-mail: clvaz@ipb.pt

Â.P. Ferreira
Polytechnic Institute of Bragança, Campus de Santa Apolónia, Apartado 1134, 5301-857, Bragança, Portugal

CISE - Electromechatronic Systems Research Centre, University of Beira Interior, Covilhã, Portugal
e-mail: apf@ipb.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_22

In a Policies Scenario taking into account both existing policies and declared intentions by countries, world primary energy demand is projected to increase by 1.2 % per year, on average, between the current year and 2035. Electricity demand is projected to grow by a higher rate, 2.2 % per year, given that it is expected that applications, formerly based on chemical energy, will be based on electrical energy in the following decades [13]. In order to cope with the worldwide climate changes, policies are being implemented to enhance the transition toward low-carbon technologies in the power sector. In this context, the share of world electricity generation from renewable sources is projected to grow, whereas the wind energy is the most representative source. According to Global Wind Energy Council (GWEC), installed wind capacity has grown to an accumulative worldwide level of 318 GW from which 35.5 GW had been installed in 2013. Europe is still the largest wind energy generator, despite the fact that other markets (e.g. USA, India, China) have also launched in recent years. Portugal accounts for about four percent of the wind energy installed capacity of the European Union, with approximately 4.7 GW of accumulated installed capacity in 2013 which is capable to generate about 20 % of electricity consumption [6].

Several factors contributed to the development of the wind energy sector in Portugal. Since 2002, the implementation of a legal stable framework by the Portuguese government and several financial support programs implemented by the European Commission have promoted the penetration of electricity generated from renewable energy sources [14]. Despite technology potential and investments in a low-carbon energy market, the progress is too slow on attending outlined targets. The main reasons for the slow progress are related with a low share of energy-related investment in R&D activities, high investments when compared with thermal based electricity, uncertain time for the return of the capital invested, technical limitations of power systems in supporting large penetration from variable renewable energy and environmental impact.

There are ten main wind farm promoters acting in the wind energy sector in Portugal, with farms connected to the transmission or distribution grid system. Each promoter is concerned, besides appropriate financial management, to ensure the maximum energy generation, with the highest availability rates and cost-effectiveness in terms of operation and maintenance. In this context, the development of performance assessment methodologies in the portfolio of a given promoter allows the identification of wind farms with the best practices in the operating stage in order to be emulated by inefficient farms. The use of DEA can contribute to enhance those methods through assessment of the potential for efficiency improvements and exploring their productivity change over time, considering the emergent interest on productivity growth in electrical utilities. This is explored by using Malmquist index which can be decomposed in efficiency change and technological change. The efficiency change can be associated with internal operating practices observed in each farm, while the technological change can be related to specific conditions in which farms have to operate, for instance, the level of wind availability in each year.

This study proposes a framework based on DEA to provide insights regarding the performance of the farms of an energy player in the Portuguese wind sector. In a first stage, DEA is used to measure the operating efficiency of the wind farms and to identify the benchmarks, followed by a second stage, were changes in wind farms productivity are investigated using the Malmquist index over 2 years, in which the wind energy sector suffered a considerable decrease in the electric energy generated. The robustness of the scores achieved by DEA models can be tested by using bootstrapping methods [17, 18]. The proposed framework is applied to a case study, giving insights into the performance assessment of wind farms from Iberwind which has a market share of 18 % on the Portuguese wind energy sector.

This study is organized as follows: next section points out a literature survey about performance assessment of wind farms, Sect. 3 presents the methodology to assess the efficiency and productivity of wind farms, Sect. 4 characterizes the context setting in the wind energy sector and applies the methodology to the case study, and finally Sect. 5 rounds up the paper with the main conclusions.

2 Literature Review

Zhou et al.[21] presented a survey on DEA energy sector and environmental modeling, from which benchmarking of electricity utilities accounted for the large number of studies although it did not include any application in wind energy sector. Regarding the methodology, this study pointed out that the constant returns to scale reference technology and the radial efficiency measures are still the most widely used specifications and there has been a growing interest on the use of Malmquist index to assess the productivity change over time.

The works performed by Iglesias et al. [12] and Pestana and Barros [16] focus on the efficiency assessment of wind farms and argue the importance to model the non-discretionary factors such as the wind speed and its availability in each farm.

Iglesias et al. [12] used DEA and Stochastic Frontier Analysis (SFA) methodologies to measure the efficiency of a group of wind farms located in Spain. Models are output oriented concerning the generated energy, based on a relationship between capital, labor and fuel, similar to a conventional energy conversion system. Capital factor is evaluated by the installed capacity in each farm and labor factor considers the number of fulltime employers responsible for operation, control and maintenance of the farms. Concerning fuel, this input is estimated based on the wind power incident per unit time on the interposed surface of the wind turbines and the annual average wind speed at each site.

Pestana and Barros [16] used SFA and stochastic production econometric frontier to assess efficiency of Portuguese wind farms from different promoters. Outputs are measured by generated energy and capacity utilization, and the inputs are price for labor and capital invested proxied by the book value of physical assets. Findings of this study are that Portuguese wind farms' operational activity is affected by heterogeneous factors such as farm size, managerial practices and ownership.

This paper improves the existing methodologies in performance assessment of wind farms from a given promoter to provide additional insights into efficiency and productivity growth over time by exploring the benchmarking analysis and Malmquist index. To increase the robustness of the efficiency and productivity results achieved, the bootstrapping framework [17, 18] is used. These methodological aspects, which have not been used in previous studies, allow a better understanding of wind farms during operating stage which can support the decision maker in benchmarking the wind farms in repowering or overpowering processes.

3 Performance Assessment Methodology

The methodology proposed in this study intends to explore the productivity and the efficiency of wind farms. In a first stage, DEA is used to assess the farms efficiency by taking into account the resources and the non-discretionary variable, the wind, available in each farm to generate electric energy. This approach enables benchmarking among farms. The robustness of efficiency scores is tested by using bootstrap framework [17]. In a second stage, we use panel data to assess the overall productivity change over time of the farms by using the Malmquist index and its components [9], efficiency change and technological change. The efficiency change measures if the farm is moving closer or farther from the frontier while the technological change measures shifts in the frontier that can be characterized by progression, regression or both. Finally, the robustness of these indexes is tested by using bootstrapping [18] which allows the identification of significant aspects that may explain the performance of each farm over time. The following sections present the proposed methodology in detail.

3.1 DEA Model

DEA is a non-parametric technique to assess the relative efficiency of an homogeneous set of Decision Making Units (DMUs) in producing multiple outputs from multiple inputs. This allows to identify the "best practices DMUs" and their linear combination defines the frontier technology. By reference to this frontier, a single summary measure of efficiency is calculated for each DMU. In the original DEA model proposed by [4], the efficiency score of each DMU is estimated by using the frontier technology characterized by Constant Returns to Scale (CRS). For an output oriented analysis, we consider a technology involving n production units defined by j ($j = 1, \ldots, n$), which use the inputs x_{ij} $(x_{1j}, \ldots, x_{mj}) \in R^m_+$, to obtain the outputs y_{rj} $(y_{1j}, \ldots, y_{sj}) \in R^s_+$, i.e., the production possibility set (PPS). In this model, the efficiency of each DMU j_o is given by the reciprocal of the factor (θ) by which the

outputs of the DMU j_o can be expanded, according to the following linear model:

$$\max \Big\{ h_{j_o} = \theta \,|\, x_{ij_o} \geq \sum_{j=1}^{n} \lambda_j \, x_{ij}, \quad i = 1, \ldots, m \tag{1}$$

$$\theta \, y_{rj_o} \leq \sum_{j=1}^{n} \lambda_j \, y_{rj}, \quad r = 1, \ldots, s$$

$$\lambda_j \geq 0, \quad \forall_j \Big\}$$

Model (1) assesses the relative efficiency of DMUs in the achievement of the output levels given the resources used. The measure of efficiency, given by $1/\theta^*$, equals to 100 % when the unit under assessment is efficient, whereas lower scores indicate the existence of inefficiencies. For the inefficient units there is evidence that it is possible to obtain higher levels of outputs with the same or lower levels of the inputs currently used. For these units, it is also possible to obtain, as by-products of the DEA efficiency assessment, a set of targets for becoming efficient. The input and output targets for a DMU j_o under assessment are obtained as follows:

$$x^o_{ij_o} = x_{ij_o} - s^*_i = \sum_{j=1}^{n} \lambda^*_j \, x_{ij}$$

$$y^o_{rj_o} = \theta^*_o \, y_{rj_o} + s^*_r = \sum_{j=1}^{n} \lambda^*_j \, y_{rj} \tag{2}$$

where the variables s^*_i and s^*_r are the slacks corresponding to the input i and output r constraints, respectively, given by the optimal solution of model (1). The benchmarks for the inefficient DMUs j_o are the units with values of $\lambda^*_j > 0$ in the optimal solution of model (1). These are the Pareto-efficient DMUs which have $\theta^*_o = 1$ and all slacks are equal to zero.

Model (1) enables to assess the Technical Efficiency (TE) for each DMU which can be due to the ineffective operation of the production process in transforming inputs into outputs and also due to the divergence of the entity from the Most Productive Scale Size (MPSS), considering the most productive frontier characterized by constant returns to scale. If a DMU has TE equals to 1, it is efficient in transforming inputs into outputs and it will have MPSS, by operating at optimal scale size. Banker et al. [1] proposed the DEA model that assesses the Pure Technical Efficiency (PTE) for each DMU by using the frontier characterized by Variable Returns to Scale (VRS) which is achieved by including the constraint $\sum_{j=1}^{n} \lambda_j = 1$ in model (1). The Pure Technical Efficiency (PTE) for each DMU enables to measure the inefficiency due to the ineffective operation of the production process in transforming inputs into outputs. The scale efficiency (SE) is measured by the distance between CRS and VRS frontiers which corresponds to the divergence

of the DMU from the MPSS and is given by the ratio $\frac{TE}{PTE}$. Thus, the Technical Efficiency (TE) is decomposed in pure technical efficiency and scale efficiency components.

For Pareto-efficient DMUs, in VRS frontier, it is possible to identify the local Returns to Scale (RTS) which enables to identify advantages in changing the scale of DMUs. In the study case under analysis, this information is very useful in repowering processes of wind farms. If increasing returns to scale hold at a Pareto-efficient DMU, then increasing its input levels by a given percentage will lead to expansion of its output levels by a larger percentage, i.e., the scale size of the DMU should be increased. If a DMU is operating at a point where decreasing returns to scale hold, it should decrease its scale size. If a DMU operates at constant returns to scale point, its scale size is considered optimal. The approach proposed in [8] is used to characterize the RTS of Pareto-efficient wind farms.

It is well known that DEA results are sensitive to sample variation which leads to deviation around the observed frontier. To overcome this uncertainty, we used bootstrapping to obtain unbiased estimates. Bootstrapping was first introduced by [7] and it is based on the idea of resampling from the given sample of observations to replicate datasets from which we can make the statistical inference. The bootstrapping approach proposed by [17] is appropriated to use with the DEA efficiency estimates which range from zero to one. For each DMU, the θ derived from model (1) is corrected for the bias to derive the bias-corrected score $\hat{\hat{\theta}}$ and the confidence interval. These scores are used to assess the wind farms performance. This procedure was implemented using the statistical software R including the FEAR library, developed by [20].

3.2 Malmquist Index on Evaluation of Overall Productivity

In energy sectors, it is of great interest the investigation of productivity change over time [21]. The Malmquist productivity index was introduced by [3] and developed further in the context of performance assessments by [9] to accomplish performance comparisons of DMUs over time. The high popularity of this method is related with several factors. Firstly, it is not necessary to use price data, assumptions of cost minimization or revenue maximization. Secondly, it can be used either in oriented or non-oriented analysis. Thirdly, it enables the determination of the total factor productivity in the generic case where production technology uses multiple inputs to produce multiple outputs by deriving efficiency scores in DEA models. Fourthly, the index is applied to the measurement of productivity change over time, and can be decomposed into an efficiency change index and a technological change index. These indexes are investigated in the case study between both years, since the wind availability decreases in 2011. It is important to know what happens with the frontier, which is captured by technological change index. In these adverse conditions, it is important to identify the behavior of each farm in catching up the

frontier, i.e., if it is getting closer or farther from the frontier, which is captured by efficiency change index.

The Malmquist index, as proposed by [9], is used to derive the overall productivity of each DMU. It is based on radial measures which are defined by distance functions. In output-oriented analysis, the output distance function is equal to the efficiency score estimated by model (1), given by $1/\theta^*$ for each DMU for a given period. Consider a set of n DMUs in period t, which use the inputs $x^t \in R_+^m$ to obtain the outputs $y^t \in R_+^s$, and the same n DMUs in period $t+1$, which use the inputs $x^{t+1} \in R_+^m$ to obtain the outputs $y^{t+1} \in R_+^s$. To simplify the notation, the efficiency score estimated for each DMU_{j_o} in period t is given by $E_o^t(t)$ while the efficiency score estimated for each DMU in period $t+1$ is given by $E_o^{t+1}(t+1)$. Thus, the score in parenthesis represents the period in each DMU is assessed while the superscript denotes the frontier technology used as reference. The Malmquist index derived for each DMU is obtained as:

$$I_o^{t+1,t} = \left(\frac{E_o^t(t+1)}{E_o^t(t)} \frac{E_o^{t+1}(t+1)}{E_o^{t+1}(t)} \right)^{\frac{1}{2}} \tag{3}$$

In terms of interpretation, a score of $I_o^{t+1,t} > 1$ indicates better performance in period $t+1$ than in period t.

The mixed-period distance functions, $E_o^t(t+1)$ and $E_o^{t+1}(t)$, can be greater, equal or lower than 1. For example, the distance function derived to the period $t+1$ for a DMU observed in period t can be lower or equal to 1 if the input-output vector of this DMU belongs to the PPS of period $t+1$. This occurs for $E_o^t(t)$ and $E_o^{t+1}(t+1)$ cases. In opposite, the distance function derived to the period $t+1$ for a DMU observed in period t is higher than 1, if the input-output vector of this DMU is outside the PPS of the period $t+1$.

According to [9], this index can be decomposed in two components: $IE_o^{t+1,t}$ and $IF_o^{t+1,t}$. The sub-index $IE_o^{t+1,t}$ corresponds to efficiency change and compares the efficiency spread between the periods observed for each DMU. The sub-index $IF_o^{t+1,t}$ corresponds to technological change and compares the relative position of the frontiers associated to periods t and $t+1$ for the input-output mix of each DMU observed. This decomposition implies that the sources of better performance can be associated with two factors: less dispersion in the efficiency score of DMU in each period and/or better productivity associated to the period frontier.

The efficiency change derived for each DMU is calculated according to:

$$IE_o^{t+1,t} = \frac{E_o^{t+1}(t+1)}{E_o^t(t)} \tag{4}$$

A value of $IE_o^{t+1,t} > 1$ means that the efficiency spread is smaller in DMU observed in period $t+1$ than the one observed in period t, measuring how much the DMU is getting closer (i.e. catching up) or farther from the frontier.

Concerning the technological change derived for each DMU, it is given by:

$$IF_o^{t+1,t} = \left(\frac{E_o^t(t)}{E_o^{t+1}(t)} \frac{E_o^t(t+1)}{E_o^{t+1}(t+1)} \right)^{\frac{1}{2}} \tag{5}$$

When $IF_o^{t+1,t}$ is higher than 1, this means that the productivity of frontier $t+1$ is better than the productivity of frontier t, which implies that the frontier has progressed. This index can be seen as an average aggregated change in technology of a DMU since it is obtained as the geometric mean of two components. The first component $(\frac{E_o^t(t)}{E_o^{t+1}(t)})$ corresponds to the distances between the frontiers t and $t+1$ when assessed for the DMU observed in period t. The second component $(\frac{E_o^t(t+1)}{E_o^{t+1}(t+1)})$ is calculated in a similar way for the same DMU observed in period $t+1$.

It is possible to analyze globally the relative position of the two frontiers, which enables to identify if the frontiers have regressed, progressed or crossed over. To do so, it is necessary to analyze each component of $IF_o^{t+1,t}$ for all DMUs observed in the periods under analysis. Some typical situations may occur: for instance, if the component is always higher than 1, this means that there has been a progression in the technology; on the other hand, if the component is always lower than 1, this means that there has been a regression in the technology; in the case when there is at least one component higher than 1 and one component lower than 1, this indicates that frontiers are crossed over, signifying that for some input-output mix the frontier progressed and for others, the frontier regressed.

The bootstrapping framework proposed by [18] is used to evaluate the robustness of the estimates of $I_o^{t+1,t}$, $IE_o^{t+1,t}$ and $IF_o^{t+1,t}$ (hereinafter I, IE and IF, respectively) obtained for each DMU, which allows the computation of confidences intervals for each index. If the interval contains the value 1, we cannot infer that significant changes occurred in the corresponding DMU. On the other hand, if the lower and upper bounds are smaller (or higher) than 1, this implies that there was a decline (or progress) in the DMU. This approach is currently used in several studies [10, 11, 15, 19]. This analysis is extended to the components of IF for all DMUs observed to find out the relative position of the frontiers.

4 Performance Assessment of Wind Farms

This section applies the methodology proposed in previous section to evaluate the performance of wind farms owned by Iberwind in Portugal. This study focus in the wind farms efficiency analysis during operating stage, for a given distribution of wind speed in the geographical location of the farms, installed capacity and number of wind turbines, oriented to the maximization of the output electric energy generated. The rationale for this context is presented in the following sub-section.

4.1 Contextual Setting

Relevant decisions and factors that affect the productivity of wind farms are prior to start-up, including for instance wind farm location and layout design, engineering design process such as the installed capacity, type of generator, turbine aerodynamics and active control system. This work focuses on the performance assessment of wind farms in the operating phase, i.e., when they perform the energy conversion and it is delivered to the utility grid. Even though the performance of a wind farm is closely linked to prior start-up phase, the operating phase is relevant throughout estimated lifetime of the assets, from the point of view of maximizing the energy generation, ensuring the highest availability rates and cost-effective operation and maintenance schemes.

Wind is a variable source of power: output rises and falls as wind strength fluctuates in a hourly or 10 min time scale, although, its variability is consistent from year to year. Wind speeds suitable for electricity generation range from approximately 5 m/s (cut in speed) to 25 or 30 m/s (cut out speed). The frequency of wind speeds usually fits a Weibull distribution and an average value, for itself, does not translate the amount of energy that a wind farm can produce.

Installed capacity and the number of wind turbines in a farm, along with the variability of wind, relate to the capacity factor of a wind farm, i.e., the ratio of actual productivity in a year to its theoretical maximum. The rated power of a unit of the wind farm (given by the ratio of the installed capacity and the number of wind turbines) if small, can lead to an higher capacity factor of the farm, and, consequently, it may not be able to produce energy at higher wind speeds, which translates in less profit. On the other hand, if the rated power of each turbine is high, it may stall at low wind speeds and the extra power at high wind speeds they are able to convert, may not compensate the higher costs of the turbine. Therefore, these resources are important to assess efficiency and productivity analysis during operating phase and may provide useful information in repowering or overpowering processes.

Concerning the output, it should be point out that electric energy generated from wind is not constrained by load demand or other market players, as currently regulated.

4.2 DEA Model

Each DMU is a wind farm which is formed by a group of wind turbines connected to the transmission or distribution grid utility. The number of DMUs under analysis is 31, spread out in North and Center of Portugal. The final data set considers 30 wind farms since one of them was eliminated due to a repowering process that began in 2010. Total capacity installed ascends to 683.75 MW through 319 wind turbines, from 15 different models, provided by five manufacturers (Vestas, Nordex, Enercon,

GE and WinWind). We consider a panel data set collected from Annual Reports and Accounts, for 2010 and 2011 years.[1] The 30 farms under analysis are located in six wind typical geographical locations in Portugal (Bragança, Vila Real, Viseu, Coimbra, Leiria and Lisboa).

The wind farms can be considered homogenous as they result from similar setup stages and use a similar generation process. The output-oriented perspective is used, as the objective of the farms is to produce maximum electric energy, taking into account the non-discretionary variable, the wind, and the resources available in each farm. For each assessed farm, the output-oriented model seeks feasible input and output levels which have the following properties: the outputs are the maximum multiple of the outputs observed in the assessed farm and the inputs are no higher than those of the assessed farm. Thus, for an inefficient farm there is evidence that it is possible to increase the level of the electrical energy produced by following the best practices observed in the benchmarks (efficient farms). These best practices can be associated with the management of the resources in each farm related to the planning of the maintenance schemes for periods in time when the wind is not suitable for producing electrical energy. Thus, the model enables to identify the inefficient farms where there is evidence that they can improve the management of their controllable resources to catch the highest level of the wind hours and, consequently, can increase the electrical energy produced.

The CRS frontier is used to assess the technical efficiency of wind farms observed. In order to model the farm activity, the input-output set should cover the full range of resources used and the outputs that are relevant for the objectives of the analysis [5]. Thus, the output corresponds to the amount of electric energy delivered to the grid and the inputs considered are the installed power, number of turbines and wind availability. The descriptive measures concerning the inputs and output under analysis are summarized in Table 1. Installed power capacity of the farms is determined by the number of wind turbines multiplied by the rated power of each one. The number of turbines relates with the area occupied by the farm. To capture the effect of the wind variability into the model, we consider the number of hours per year that wind speed is within the range defined by cut in and cut out speeds (hereinafter named wind hours). For each wind farm location, the wind data is collected from a meteorological data base throughout identification of the station which represents its wind profile, defined by the nearest meteorological station. The inclusion of this non-discretionary input assures that a farm with unfavorable conditions regarding wind resource is not penalized in the performance assessment. The wind hours is an internal non-discretionary input which should be used for the definition of the PPS, according to [2]. Data concerning the maintenance schemes and operation costs are confidential and, consequently, they are not included in the model.

[1]The constraint of the panel data is limited to 2010 and 2011, because there is no available wind data from meteorological stations in former years and also in recent years.

Table 1 Mean and standard deviation values for inputs and output of wind farms

	2010		2011	
	Mean	Sta. Dev.	Mean	Sta. Dev.
Inputs				
Installed power (MW)	22.4	30.4	22.4	30.4
No of wind turbines	10.4	10.7	10.4	10.7
Wind hours	3773.6	1082.4	3280.5	980.4
Output				
Electric Energy (GWh)	56.5	81.5	51.1	75.8

Table 2 Summary results of original and bootstrapped efficiency scores

	Year 2010			Year 2011		
	Bias cor. eff. (%)	Bias (%)	Eff. est. (%)	Bias cor. eff. (%)	Bias (%)	Eff. est. (%)
Mean	72.92	−8.00	77.73	66.33	−10.54	73.97
Sta. Dev.	10.13	4.30	12.33	10.86	4.92	13.44

The standard deviation of observed variables is quite high compared with the mean values, indicating a considerable amount of diversity in the wind farms.

The summary of the technical efficiency estimates, using the formulation shown in model (1) are presented in Table 2. The robustness of these estimates is tested by calculating the bias-corrected efficiency scores (as the inverse of $\hat{\hat{\theta}}$) [17] which summary results are also presented in Table 2.

The efficiency estimated scores are relative, since the farms in a given year are only compared with all farms in the sample operating in the same year. We may observe that the wind farms under analysis are more homogenous in 2010 than in 2011 which is confirmed by bootstrapping analysis. The absolute value of bias is slightly higher in 2011 due to the differences between bias-corrected efficiency scores, and efficiency estimates are higher in the same period. Globally, this indicates that farms moved farther from the frontier. This effect is captured by the analysis of efficiency change index (*IE*) for each farm, which is explored in productivity analysis. Figure 1 presents the average of bias-corrected efficiency of the farms located in the same region for both years which indicates that the level of efficiency spread increased in 2011 for all regions.

4.3 Benchmarking Analysis

The benchmark farms and their best practices should be identified in order to be emulated by inefficient units. These practices may be related to the use of more efficient wind turbines, enhanced wind farm design and layout, better operation and maintenance schemes, which may be used to support the inefficient farms to achieve

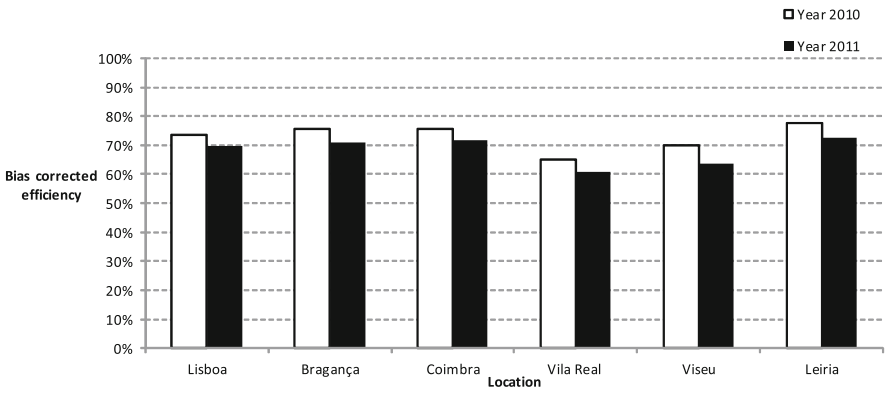


Fig. 1 Comparison of the average of bias-corrected efficiency of the farms located in the same region for 2010 and 2011 years

the appropriate targets. It is also important to identify the nature of returns to scale of the Pareto-efficient farms to explore changes in their size.

From the sample used, there are only 3 efficient farms: Achada, Candeeiros and Pampilhosa. These farms maintain the efficiency status in both years. In 2011, Achada and Candeeiros are the benchmarks, being used as reference 27 and 22 times, respectively. There are no units which are compared with Pampilhosa, since this farm is the largest unit in terms of number of wind turbines and installed capacity. In the following, we explore the profile of the benchmarks in terms of location and type of wind turbines used.

Benchmarks are located in areas with high wind potential (Lisboa, Leiria and Coimbra) and their energy conversion system is based on asynchronous generators. The wind turbines of Achada are from Nordex manufacturer while the wind turbines of Candeeiros and Pampilhosa are from Vestas. These farms are the largest ones while Achada is a smaller farm. Figure 2 compares the age, inputs and output of benchmarks with those observed in inefficient farms, in 2011 (the some profile occurs in 2010). In this graph the scores were normalized by the average scores observed in benchmarks to simplify the comparison. The installed capacity of inefficient units is, on average, 80 % less of that observed in benchmarks and the electric energy generated follows a decrement of the same magnitude. The inefficient units have, on average, 67 % less number of wind turbines of those observed in benchmarks. Given that wind hours in geographical areas where inefficient units are located have a small reduction (about 14 %), this suggests inefficient farms are prone to an overpowering process, in order to increase their output, as they are not exploring all wind energy potential. The fact that inefficient farms are, on average, 15 % older than benchmarks, may explain some inefficiency in some farms.

In both years, the most inefficient unit is the same farm: the Lomba Seixa I with scores equal to 56.7 %, in 2010, and 49.4 %, in 2011. The lowest score can be due

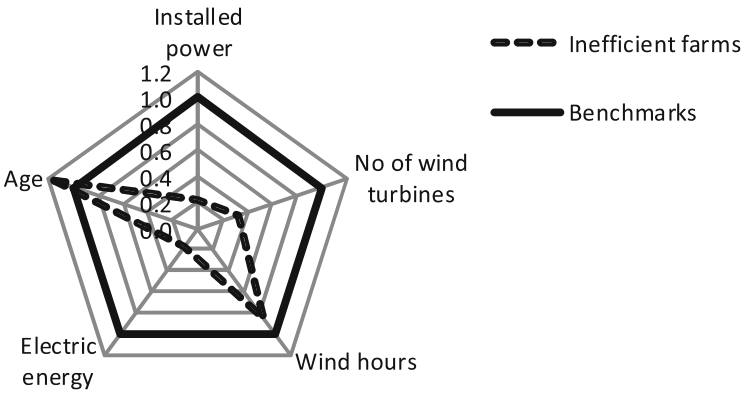


Fig. 2 Comparison between benchmarks and inefficient farms in 2011

to age of technology of the energy conversion system installed in this farm, as it is 11 years old.

The results also indicate that all inefficient farms have slack in the constraint relative to number of wind turbines. Conversely, there are no farms with slack in installed capacity. Thus, the inefficient farms would increase the energy generated by using a lower number of wind turbines with higher rated power, providing the same installed power. This upshot is important in repowering processes. Although, it is important to use a relevant number of turbines to catch the wind potential in a given location, these results suggest that wind farm design could be enhanced and used to decrease the environment impact of future wind farms projects.

These findings should be explored and discussed with the promoter, in order to enhance performance of the wind farms. For inefficient units, it is possible to specify appropriate targets based on internal benchmarking, as proposed in the next section.

4.3.1 Target setting

For each inefficient farm, we can define targets for performance improvement. These targets are determined by linear combination of the benchmarks for each inefficient unit. For example, the technical efficiency of farm 17 (Lousã I) is about 67 %. The scores for 2011 period, regarding inputs and output of this farm, DEA targets (determined by (2)) and peers, are presented in Table 3.

The target for a given variable (input or output) of farm 17 is defined by the linear combination of 0.213 of the score observed in farm 1 (Achada) and 0.302 of the score observed in farm 7 (Candeeiros). Farm 17 is larger than farm 1 and smaller than farm 7. Targets indicate that it is possible to increase the electric energy to 107.1 GWh by using the actual installed capacity with the same wind hours available in 2011, with a slack roughly equal to 2 turbines. In theory, the ratio between the total installed power and the number of turbines should be increased,

Table 3 Target setting of Farm 17

			Benchmarks	
			Farm 1	Farm 7
	Observed	Target	$\lambda = 0.213$	$\lambda = 0.302$
Installed power (MW)	35	35	6.9	111
No of wind turbines	14	11.8	3	37
Wind hours	2598	2598	5397	4787
Electric energy generated (GWh)	71.7	107.1	22.4	338.9

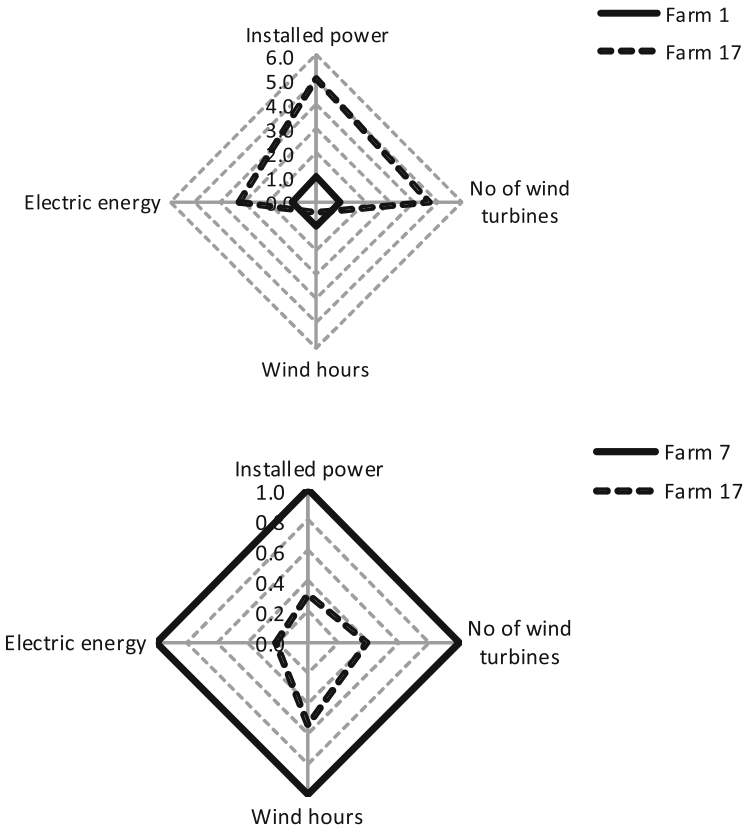


Fig. 3 Comparison between actual values of Farm 17 with benchmarks 1 and 7

via increment of the rated power of each turbine. We can compare the actual inputs and output observed in farm 17 with each benchmark by using the radar graphs in Fig. 3, where the scores were normalized by those observed in benchmarks to simplify the comparison. Farm 17 has an installed power and a number of wind turbines which are almost 5 times higher, but 48 % less wind hours than those observed in benchmark 1. Taking into consideration the exogenous characteristic

of the input wind hours, it is not evident a possible increase in electric energy production. On the other hand, farm 17 has an installed power and a number of wind turbines about 70 % and 60 %, respectively, less than those observed in benchmark 7 while the reduction in the wind hours is about 50 %. We can conclude that inputs of farm 17 are, on average, 60 % lower than those observed in benchmark 7 and a similar percentage of reduction in generated electric energy would be expected and not a decrease of 80 %, as observed. From the comparison with peers, namely wind farm 7, the farm 17 could produce higher level of electric energy from the resources observed. Hence, it is necessary to identify the best practices observed in benchmarks 1 and 7 which should be emulated by inefficient farm 17.

As the production technology of wind farms is characterized by constant returns to scale, the farm efficiency score, $1/\theta^*$, includes sources related to the inefficient operation and scale size. Next, we explore the scale size of the farms based on internal benchmarking.

4.3.2 Exploring Changes in Wind Farms Size

As the scale size affects the productivity of a DMU, it is important to calculate the scale efficiency to measure the distance between CRS and VRS frontiers at the scale size of the assessed unit. So, the larger the difference between TE and PTE efficiency scores, the lower the value of scale efficiency is, and the adverse impact of scale size on productivity is more significative. The average scale efficiency score is, on average, 95.38 %, and 93.31 %, in 2010 and 2011, respectively. This means that scale size only affects the productivity of a small proportion of units observed (Jarmeleira, Borninhos, Rabaçal, Chiqueiro, Malhadizes, Degracias, Lousã I, Lousã II, Malhadas), where the scale efficiency has the lowest scores, with a range between 73.4 % and 88.9 %. This strengthens the use of constant returns to scale frontier technology to assess the wind farms efficiency.

The analysis of local returns to scale according to [8] shows that Achada, Candeeiros and Pampilhosa are characterized by an optimum size. Jarmeleira, Lousã II, Malhadas and Rabaçal have increasing returns to scale, which indicates that the size of these units could be increased with a repowering process which enables increasing their productivity. There is no unit which has decreasing returns to scale and, consequently, there is no one with higher size than the required, taken into account the level of electric energy generated.

4.4 Productivity Analysis

In a second stage, we investigate the productivity of wind farms by disentangling the efficiency change and technological change effects observed in wind farms in 2010 and 2011. An aggregate analysis is performed by identifying the global effects which had occurred in the period under analysis. Changes in efficiency

Table 4 Significant scores for *I*, *IE* and *IF*

	I	IE	IF
Improvement	1	1	–
Deterioration	27	19	17
Stagnation	2	10	13

(*IE*), technology (*IF*) and productivity (*I*) indexes of farms are explored through identification of scores higher, lower or equal to 1 which correspond to improvement, deterioration or stagnation, respectively. This analysis is complemented with bootstrapping framework, as proposed by [18], to identify if those changes, for each farm, are significant. Table 4 aggregates the significant results in terms of number of wind farms which improve, decline or maintain the performance for each index.

We observed that 27 farms decreased overall productivity levels in year 2011, as indicated by significant scores of *I* index. This effect is mainly due to deterioration in the productivity levels of the frontier for some inputs-output mix and decreasing efficiency levels in some farms. Only Serra Escusa improves overall productivity level due to improvement on its efficiency in 2011. Pampilhosa and Candeeiros maintain overall productivity levels in 2011.

There are 19 farms that moved farther from the frontier in 2011, as indicated by significant scores of *IE* index. These farms had the worst performance in 2011, so the reasons for that should be investigated. Only Serra Escusa moved closer to the best practices. It is recommended to identify how this farm carried out its operations and maintenance services in order to be emulated by the inefficient farms. The remaining farms maintained the efficiency spread levels observed in 2010.

Globally, the productivity of the best-practices frontier decreased considerably in 2011 for the input-output combinations of 17 farms, although for the remaining input-output mix, the frontier maintained the level of productivity observed in 2010. This is connected with the reduction of electric energy generated observed in wind energy sector in 2011. Next, we explore the relative position of frontiers for the farms observed in each period. Thus, we analyze if the ratios of *IF* ($\frac{E^{2010}(2011)}{E^{2011}(2011)}$, $\frac{E^{2010}(2010)}{E^{2011}(2010)}$) are statistical significant throughout bootstrap framework [18]. Table 5 aggregates the significant results in terms of number of wind farms which improve, decline or maintain the performance for each ratio.

The inputs-output combinations of 14 farms observed in 2010 are located in areas of the PPS where the productivity of the frontier declined. The remaining farms are located in areas of the PPS where the frontier maintained the productivity. During 2011, there are 18 inputs-output combinations of wind farms located in areas of the PPS where the frontier regressed, while the other remaining farms are located in areas where the frontier maintained the productivity. There is no statistical evidence of crossed frontiers for all input-output combinations.

Table 5 Significant scores for ratios of *IF*

	$E^{2010}(2010)/E^{2011}(2010)$	$E^{2010}(2011)/E^{2011}(2011)$
Improvement	0	0
Deterioration	14	18
Stagnation	16	12

5 Conclusions

This study proposes a methodology to assess the efficiency and productivity change of wind farms, which can support decision makers during operating phase of wind farms, in repowering processes and also in project design and layout of new farms. In a first stage, the efficiency assessment of wind farms enables the identification of benchmark profiles, setting targets for inefficient units and also the exploitation of the scale size of existing farms. The second stage explores the efficiency and productivity over time of wind farms by identifying the global effects which occurred in terms of changes in internal practices observed and productivity of the frontier, during the period under analysis. These findings correspond to the additional insights regarding the efficiency and productivity assessment of wind farms which make this paper different from the previous studies.

Regarding the operating stage of the farms analyzed, 3 farms are the benchmarks, whose best practices can be related to the well-performing operations and maintenance programs. Between 2010 and 2011, different profiles of wind farms were identified in terms of overall productivity change, efficiency change and technological change. Almost all farms decreased overall productivity levels, mainly due to the decline in the productivity levels of the frontier, which is in accordance with the decrease in wind availability, measured in wind hours, observed in 2011. The productivity of the frontier declined for some input-output combinations observed in 2011 and for the other combinations, the frontier maintained its productivity. In the later case, there is one farm that improved its overall productivity due to the improvement of its efficiency in 2011 and two farms which maintained overall productivity as they kept the efficiency levels. We observed also that 19 farms had the worst performance in 2011 which requires further investigation to reveal the reasons.

Further research should be conducted using a larger panel data set in order to analyze the impact of wind availability on the productivity of wind farms. The inclusion of variables concerning the operation and maintenance schemes should also be explored in future performance assessments of wind farms.

References

1. Banker, R.D., Charnes, A., Cooper, W.W.: Some models for estimating technical and scale inefficiencies in Data Envelopment Analysis. Manag. Sci. **30**(9), 1078–1092 (1984)
2. Camanho, A.S., Portela, M.C., Vaz, C.B.: Efficiency analysis accounting for internal and external non-discretionary factors. Comput. Oper. Res. **36**(5), 1591–1601 (2009)
3. Caves, D.W., Christensen, L.R., Diewert, W.E.: The economic theory of index numbers and the measurement of input, output and productivity. Econometrica **50**, 1393–1414 (1982)
4. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring efficiency of decision-making units. Eur. J. Oper. Res. **2**(6), 429–444 (1978)
5. Dyson, R.G., Allen, R., Camanho, A.S., Podinovski, V.V., Sarrico, C.S., Shale, E. A.: Pitfalls and protocols in DEA. Eur. J. Oper. Res. **132**, 245–259 (2001)
6. Direção Geral de Energia e Geologia (DGEG): Renováveis, Estatísticas rápidas (Nov. 2013). Available from: http://www.dgeg.pt/
7. Efron, B.: Bootstrap method: another look at the jackknife. Ann. Stat. **7**(1), 1–26 (1979)
8. Färe, R., Grosskopf, S., Lovell, C.A.K.: The Measurement of Efficiency of Production. Kluwer, Boston (1985)
9. Färe, R., Grosskopf, S., Lindgren, B., Roos, P.: Productivity developments in swedish hospitals: a Malmquist output index approach. In: Charnes, A., Cooper, W.W., Lewin, A., Seiford, L. (eds.) Data Envelopment Analysis: Theory, Methodology and Applications, pp. 253–272. Kluwer, Boston (1994)
10. Gilbert, R.A., Wilson, P.W.: Effects of deregulation on the productivity of Korean banks. J. Econ. Bus. **50**(2), 133–155 (1998)
11. Horta, I.M., Camanho, A.S., Johnes, J., Johnes, G.: Performance trends in the construction industry worldwide: an overview of the turn of the century. J. Product. Anal. **39**(1), 89–99 (2012)
12. Iglesias, G., Castellanos, P., Seijas, A.: Measurement of productive efficiency with frontier methods: a case study for wind farms. Energy Econ. **32**(5), 1199–1208 (2010)
13. International Energy Agency (IEA): World Energy Outlook 2010. IEA Publications, Paris (2010)
14. Martins, A.C., Marques, R.C., Cruz, C.O.: Public-private partnerships for wind power generation: the Portuguese case. Energy Policy **39**, 94–104 (2011)
15. Odeck, J.: Statistical precision of DEA and malmquist indices: a bootstrap application to norwegian grain producers. Omega **37**(5), 1007–1017 (2009)
16. Pestana Barros, C., Sequeira Antunes, O.: Performance assessment of portuguese wind farms: ownership and managerial efficiency. Energy Policy **39**(6), 3055–3063 (2011)
17. Simar, L., Wilson, P.W.: Sensitivity analysis of efficiency scores: how to bootstrap in nonparametric frontier models. Manag. Sci. **44**(1), 49–61 (1998)
18. Simar, L., Wilson, P.W.: Estimating and bootstrapping Malmquist indices. Eur. J. Oper. Res. **115**, 459–471 (1999)
19. Tortosaausina, E., Grifelltatje, E., Armero, C., Conesa, D.: Sensitivity analysis of efficiency and malmquist productivity indices: an application to spanish savings banks. Eur. J. Oper. Res. **184**(3), 1062–1084 (2008)
20. Wilson P.W.: FEAR: a software package for frontier efficiency analysis with R. Socio-Econ. Plan. Sci. **42**(4), 247–254 (2008)
21. Zhou, P., Ang, B.W., Poh, K.L.: A survey of data envelopment analysis in energy and environmental studies. Eur. J. Oper. Res. **189**(1), 1–18 (2008)

Multi-period and Multi-product Inventory Management Model with Lateral Transshipments

Joaquim Jorge Vicente, Susana Relvas, and Ana Paula Barbosa Póvoa

Abstract Inventory management plays an important role in supply chains. Through a correct inventory management policy, supply chains can close the gap created by the imbalance between supply and demand, eliminating costly supply chains. This paper aims to contribute to this goal and presents an Inventory Management (IM) policy implemented on a Mixed Integer Linear Programming (MILP) model that optimizes the flow of products through a multi-period and multi-product supply chain. Normally distributed demands are received at the retailers who replenish their stock from the regional warehouses, which, in turn, are supplied by a central warehouse. Lateral transshipment is allowed among regional warehouses and among retailers. In order to validate and compare the proposed policy against commonly used policies, the Continuous Review and the Periodic Review policies are modeled using the same approach and acting over the same system. The comparison of inventory management policies shows that the IM policy outperforms the classical policies in terms of material availability leading to an overall reduction of operational costs.

1 Introduction

Supply chain management deals with the organization of the flows of products and information throughout the supply chain structure so as to ensure the requirements of customers, while minimizing operating costs. To attain such goal inventory management is, within supply chains, a key activity since it ensures the continuity and balance between supply and demand. Beamon [3] characterized inventory management through the different activities in the chain and enhanced the importance of developing correct inventory management policies. In order to avoid such problem there is a need of an integrated approach for the planning and control of inventory throughout the entire supply chain. In this context inventory management appears

J.J. Vicente (✉) • S. Relvas • A.P. Barbosa Póvoa
CEG-IST Centre for Management Studies, Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais, 1049-001 Lisboa, Portugal
e-mail: joaquim.jorge.vicente@gmail.com; susana.relvas@tecnico.ulisboa.pt; apovoa@tecnico.ulisboa.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_23

as an important and challenging problem, since decisions made by a chain member may affect, with different impacts, the remaining supply chain entities.

Krautter [6] presented new perspectives for corporative management on the inventory theory area and stated that inventories appear as the result of mismanagement of the different supply chain processes. The need to fully control the processes was identified by the author as a way of optimizing inventories. Giannoccaro and Pontrandolfo [4] studied the integration and coordination of inventory policies adopted by different supply chain actors so as to smooth material flow and minimize costs while responsively meeting customer demand.

Later on, [2] proposed a method to determine control parameters on a one-warehouse/N-retailer network. An approximate optimization of reorder points for a continuous review installation stock (R, Q) policy was considered in a two-echelon distribution inventory system with stochastic demand. All orders are placed at the retailers and the retailers replenish their stock from the warehouse that, in turn, replenishes its stock from an outside supplier with infinite supply.

Abdul-Jalbar et al. [1] dealt with the classic deterministic one-warehouse multi-retailer inventory/distribution system where customer demand rates were assumed to be known and constant and there was no backlog or lost sales. The retailers placed orders to satisfy customer demands generating demands at the warehouse. Order lead times were assumed to be instantaneous, so no lead time was considered. Costs at each facility consisted of a fixed charge per order and of a holding cost.

Ozdemir et al. [8] studied the multi-location transshipment problem with capacitated transportation. They used a simulation based approach that incorporates transportation capacity such that transshipment quantities between stocking locations are bounded due to transportation media capacity or the location's transshipment policy.

Hsiao [5] investigated the classic deterministic one-warehouse multi-retailer inventory/distribution system. In this study the customer demand rates were assumed to be known and constant. Shortages were not permitted and lead times were assumed negligible. A method that reached the optimal solution in most of the instances studied was proposed.

In the same year, [7] also developed an inventory control system for a one-warehouse multiple-retailers supply chain. They considered only one product. A mixed integer linear model was proposed to determine the optimal inventory and distribution plan that minimized total related costs. The efficiency of the inventory control system was compared to a periodic review policy.

More recently, [9] researched and reviewed inventory models with lateral transshipments. Models of many different systems have been considered. This paper provides a literature review which categorizes the research to date on lateral transshipments, so that these differences can be understood and gaps within the literature can be identified. Yousuk and Luong [10] present a model of two-retailer inventory system with preventive lateral transshipment. Each retailer employs base stock periodic review policy.

Most of these studies dealt with a supply chain management structure formed by a single warehouse that supplies multiple retailers with a single product item. A single period of analysis also characterizes most of the models proposed. Based

on this analysis it is clear that some research space still exists to generalize the inventory management models proposed. In particular, new models should be developed to deal with more generic supply chain structures. Aspects that will allow a closer description of the real problems should be explored namely: (i) generic structure with links between the different entities present (e.g. transshipment); (ii) inclusion of supply lead times; (iii) safety stock considerations and finally (iv) inclusion of lost sales at all retailers.

The present work follows this need and develops an Inventory Management (IM) policy that may support in an optimized way, by minimizing total operational costs, the definition of the product flows through a multi-warehouse/multi-retailer/multi-product and multi-period supply chain. Costs include ordering, holding in stock and in transit, transportation, transshipping and lost sales. The system in study and the associated IM policy are modeled through a MILP model. The classical Continuous Review (CR) and the Periodic Review (PR) policies are also modeled by means of mathematical programming models that act in the same system, so as to compare the results of the three studied policies.

The remainder of this paper is organized as follows. The problem definition is given in Sect. 2. Section 3 describes the IM mathematical model and the CR and PR mathematical models are presented respectively in Sects. 4 and 5. An inventory management case study is presented in Sect. 6. Finally the conclusions are drawn in Sect. 7.

2 Problem Definition

A generic supply chain is considered in this current study. It comprises multiple regional warehouses and multiple retailers as depicted in Fig. 1, where multiple products are distributed over a given time horizon.

The structure considered assumes that retailers replenish their inventories from the regional warehouses, these replenish their inventories from a central warehouse and customer demand is observed at the retailers. Each retailer has a normally

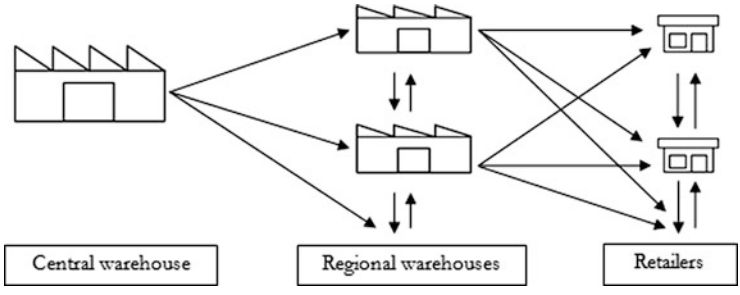


Fig. 1 Supply chain structure

distributed demand with a mean and a standard deviation in each unit of time, which is independent of demands of the other retailers. It is also assumed that storage and transportation capacities, in regional warehouses and retailers, are limited and that transportation occurs after orders have been placed. Also the storage and transportation capacity from the central warehouse is unlimited.

Lateral transshipment between regional warehouses or between retailers is allowed. If the demand in a given time period and at a given retailer is not satisfied, this is considered as a lost sale. Different cost types are included. These are related to the ordering process, holding in storage and holding in-transit, transportation, transshipping and lost sale. Fixed ordering costs occur each time a regional warehouse or a retailer places an order and are related to the ordering activity, being independent of the quantity ordered. Holding costs are defined for both storage and in-transit inventory. The first ones are defined per unit stored and per time period on each regional warehouse or retailer. The second ones are defined per unit of product transported and are dependent of the lead times. Transportation costs are considered per unit of material transported between the different stages of the supply chain. Related to these are the transshipment costs that represent the costs of transporting items between two locations belonging to the same echelon. Finally, lost sale costs are associated to the demand that cannot be satisfied and are defined per unit of product. Thus, it is important to effectively represent and optimize the flows of products through the entire supply chain so as to minimize costs. These aspects are included in the problem under study and the relevant decision that needs to be modeled is then to determine the shipping quantity to be sent from the regional warehouses to each retailer in each time period so as to minimize the total system costs. The problem in study can be generally defined as follows:

Given:

- The planning time horizon and a defined discrete time scale;
- The number of regional warehouses and retailers;
- The number of products;
- Initial inventory level of each product at each regional warehouse and retailer;
- Customer demands for each product in all time periods;
- Storage capacities in each regional warehouse and retailer per time period;
- Transportation capacities between entities;
- Safety stock by product in each regional warehouse and retailer;
- Transportation lead times between entities;
- Ordering costs per order of each product at each regional warehouse and retailer (independent of order quantity);
- Unitary holding cost per time period per product at each regional warehouse or retailer;
- Unitary holding cost per time period per product in transit (dependent of lead time);
- Unitary transportation and transshipping cost per product;
- Unitary lost sale cost per product and per each time period.

Determine:

- The inventory profiles for each product throughout the time horizon in each regional warehouse and retailer;
- The flows of products across the supply chain for each time period. These involve shipping quantities between entities on different supply chain levels and transshipment quantities between entities on the same supply chain level;
- The lost sale quantities of each product at each retailer at each time period.

So as to minimize the total costs over the time horizon considered. A mathematical programming formulation is proposed for the problem, which will be presented in the subsequent section.

3 Inventory Management Model (IM Model)

The supply chain inventory management problem presented is formulated as a MILP model. As referred, it aims to minimize the total costs during the time horizon in study. The MILP model considers time through a discretized time scale, where the time intervals have equal durations.

The indices, sets, parameters and variables (binary and continuous) used in the model formulation are defined using the following notation:

Indices

i product
j, k, l, m entity node
t time period

Sets

$i \in P = \{1, 2, \ldots, NP\}$ products
$j, k, l, m \in I = \{0, 1, 2, \ldots, NW, NW+1, NW+2, \ldots, NW+NR\}$ supply chain nodes
$t \in T = \{1, 2, \ldots, NT\}$ time periods
$W = \{1, 2, \ldots, NW\}, W \subset I$ regional warehouses
$R = \{1, 2, \ldots, NR\}, R \subset I$ retailers
$W_o = \{0\}, W_o \subset I$ central warehouse
$DN = \{1, 2, \ldots, NW, NW+1, NW+2, \ldots, NW+NR\}, DN \subset I$ demand nodes (regional warehouses and retailers)
$SN = \{0, 1, 2, \ldots, NW\}, SN \subset I$ supply nodes (central warehouse and regional warehouses)
Note that $W_o \cup W \cup R = I$

Parameters

BGM a large positive number
CD_{ijt} customer demand of the product i at node $j \in R$ in time period t (note that

customer demand occurs only at the retailers, but not at the warehouses)
HOC_{ij} unitary holding cost of the product i at node $j \in DN$ per time period
$HTCR_{ijk}$ unitary holding cost of the product i in transit from a regional warehouse $j \in W$ to a retailer $k \in R$ per time period
$HTCW_{ijk}$ unitary holding cost of the product i in transit from a central warehouse $j \in W_o$ to a regional warehouse $k \in W$ by time period
Ito_{ij} initial inventory level of the product i at node $j \in DN$
LSC_{ijt} unitary lost sale cost of the product i at node $j \in R$ in time period t
LTT_{jk} transportation lead time from node $j \in I$ to node $k \in I$
OC_{ij} ordering cost of the product i at node $j \in DN$ (note that ordering cost is independent of quantity of product i)
SS_{ij} safety stock of the product i at node $j \in DN$
STC_{jt} storage capacity at node $j \in DN$ in time period t
$TRACMAX_{jk}$ maximum transportation capacity from node $j \in DN$ to node $k \in DN$
$TRACMIN_{jk}$ minimum transportation capacity from node $j \in DN$ to node $k \in DN$
$TRCR_{ijk}$ unitary transportation cost of product i from a regional warehouse $j \in W$ to a retailer $k \in R$
$TRCW_{ijk}$ unitary transportation cost of product i from a central warehouse $j \in W_o$ to a regional warehouse $k \in W$
$TSCW_{ijl}$ unitary transshipment cost of product i from a regional warehouse $j \in W$ to another regional warehouse $l \in W$
$TSCR_{ikm}$ unitary transshipment cost of product i from a retailer $k \in R$ to another retailer $m \in R$

Continuous variables

FI_{ijt} inventory of product i at node $j \in DN$ at the end of time period t
LS_{ijt} lost sales of product i at node $j \in R$ during time period t (note that lost sales only occur at the retailers)
SQ_{ijkt} shipping quantity of product i from node $j \in I$ to node $k \in I$ during time period t

Binary variable

$BV1_{ijt}$ equal to 1 if an order of product i is placed by node $j \in DN$ in time period t; 0 otherwise

Objective function

The objective function consists of the minimization of the total cost is given as follows:

$$\text{minimize total cost} = \sum_{i\in P}\sum_{j\in DN}\sum_{t\in T} OC_{ij} \times BV1_{ijt} + \sum_{i\in P}\sum_{j\in DN}\sum_{t\in T} HOC_{ij} \times FI_{ijt}$$

$$+\sum_{i\in P}\sum_{j\in W_o}\sum_{k\in W}\sum_{t\in T} HTCW_{ijk}\times LTT_{jk}\times SQ_{ijkt}+\sum_{i\in P}\sum_{j\in W}\sum_{k\in R}\sum_{t\in T} HTCR_{ijk}\times LTT_{jk}\times SQ_{ijkt}$$

$$+\sum_{i\in P}\sum_{j\in W_o}\sum_{k\in W}\sum_{t\in T} TRCW_{ijk}\times SQ_{ijkt}+\sum_{i\in P}\sum_{j\in W}\sum_{k\in R}\sum_{t\in T} TRCR_{ijk}\times SQ_{ijkt}$$

$$+\sum_{i\in P}\sum_{j\in W}\sum_{l\in W}\sum_{t\in T} TSCW_{ijl}\times SQ_{ijlt}+\sum_{i\in P}\sum_{k\in R}\sum_{m\in R}\sum_{t\in T} TSCR_{ikm}\times SQ_{ikmt}$$

$$+\sum_{i\in P}\sum_{j\in R}\sum_{t\in T} LSC_{ijt}\times LS_{ijt} \tag{1}$$

The first term of objective function (1) is the ordering cost. The second term expresses the holding costs at both stages of the supply chain, regional warehouses and retailers. The third and the fourth terms are the holding cost in transit at both supply chain stages. The holding cost in transit is lead time dependent. The fifth and the sixth terms are the transportation costs at both supply chain stages. The transshipping cost, between regional warehouses and between retailers, is given by the seventh and eighth terms and, finally, the last term represents the lost sale cost.

Constraints

The model developed consists of different types of constraints. These are grouped into: inventory constraints; shipping constraints; storage capacities; transportation capacities; safety stock policy and non-negativity and binary conditions.

Inventory constraints

Inventory constraints have to be defined for both the regional warehouses and retailers, taking into account all inputs and outputs at each time period.

Regional warehouses:

The total incoming quantity at each regional warehouse is equal to the shipping quantity from the central warehouse to the regional warehouse, plus the transshipping quantity from the others regional warehouses, considering the transportation lead time through the introduction of a time lag.

The total outgoing quantity at each regional warehouse is equal to the sum of shipping quantity from the regional warehouse to the retailers plus the transshipping quantity to the others regional warehouses, at time period t. For $t = 1$ the inventory of product i at the end of this time period at regional warehouses is given by constraints (2), which takes into account the initial inventory level at each regional warehouse Ito_{im}.

$$FI_{imt} = Ito_{im} + SQ_{i,0,m,t-LTT_{0m}|_{LTT_{0m}=0}} - \sum_{k\in R} SQ_{imkt} - \sum_{l\in W\wedge l\neq m} SQ_{imlt}$$

$$+\sum_{l\in W\wedge l\neq m} SQ_{i,l,m,t-LTT_{lm}|_{LTT_{lm}=0}}, \quad i\in P, m\in W, t=1 \tag{2}$$

For the remaining time periods the inventory at the end of these time periods at regional warehouses is given by constraints (3).

$$FI_{imt} = FI_{i,m,t-1} + SQ_{i,0,m,t-LTT_{0m}}|_{LTT_{0m}<t} - \sum_{k\in R} SQ_{imkt} - \sum_{l\in W\wedge l\neq m} SQ_{imlt}$$

$$+ \sum_{l\in W\wedge l\neq m} SQ_{i,l,m,t-LTT_{lm}}|_{LTT_{lm}<t}, \quad i \in P, m \in W, t \in T\backslash\{1\} \tag{3}$$

Retailers:

At each retailer, the incoming quantity is equal to the sum of the shipping quantity from the regional warehouses to that retailer, plus the transshipping quantity from the others retailers, at time period t, considering the transportation lead time.

At each retailer the outgoing quantity is equal to the customer demand minus the lost sale of that retailer plus the transshipping quantity to the others retailers, at time period t.

For $t = 1$ the inventory of product i at the end of this time period at the retailers is given by constraints (4), which accounts for the initial inventory of product i at retailer k (Ito_{ik}) whereas constraints (5) is applicable for the remaining time periods.

$$FI_{ikt} = Ito_{ik} + \sum_{j\in W} SQ_{i,j,k,t-LTT_{jk}}|_{LTT_{jk}=0} - (CD_{ikt} - LS_{ikt}) - \sum_{m\in R\wedge m\neq k} SQ_{ikmt}$$

$$+ \sum_{m\in R\wedge m\neq k} SQ_{i,m,k,t-LTT_{mk}}|_{LTT_{mk}=0}, \quad i \in P, k \in R, t = 1 \tag{4}$$

$$FI_{ikt} = FI_{i,k,t-1} + \sum_{j\in W} SQ_{i,j,k,t-LTT_{jk}}|_{LTT_{jk}<t} - (CD_{ikt} - LS_{ikt}) - \sum_{m\in R\wedge m\neq k} SQ_{ikmt}$$

$$+ \sum_{m\in R\wedge m\neq k} SQ_{i,m,k,t-LTT_{mk}}|_{LTT_{mk}<t}, \quad i \in P, k \in R, t \in T\backslash\{1\} \tag{5}$$

Shipping constraints

Since transportation occurs after an order has been placed from a destination to its source, it is assumed that the fixed ordering cost is always incurred when the transportation occurs. Hence, if the transportation amount is not zero the binary variable $BV1_{ikt}$ equals 1, as implied in constraints (6). The left hand side of this constraint represents the quantity received by a regional warehouse, which can come from the central warehouse (first term) or any other regional warehouse (second term).

$$SQ_{i0kt} + \sum_{l\in W\wedge l\neq k} SQ_{ilkt} \leq BGM \times BV1_{ikt}, \quad i \in P, k \in W, t \in T \tag{6}$$

Equivalents constraints are defined for retailers, constraints (7). The BGM value will have a value that is valid as an upper bound of any quantity that can be ordered

by a regional warehouse or retailer.

$$\sum_{j \in W} SQ_{ijkt} + \sum_{l \in R \wedge l \neq k} SQ_{ilkt} \leq BGM \times BV1_{ikt}, \quad i \in P, k \in R, t \in T \tag{7}$$

Storage capacities

The total inventory stored at any node, given by the sum of the inventory level of each product i must respect the storage capacity in each demand node j at any time period t, which is enforced by constraints (8). In this formulation we consider storage capacities time dependent to illustrate that, depending on the time period, capacities may vary since we are dealing on an operational perspective. Also the storage capacity of the central warehouse is unlimited.

$$\sum_{i \in P} FI_{ijt} \leq STC_{jt}, \quad j \in DN, t \in T \tag{8}$$

Transportation capacities

At any time period t, the sum of the shipping quantity of each product i must respect the transportation lower and maximum limits between each two nodes j and k, as stated in constraints (9) and (10). Note that the quantities in transit were not considered for the usage of transportation capacity, but for the total transportation capacity. Also the transportation capacity from the central warehouse is unlimited.

$$\sum_{i \in P} SQ_{ijkt} \leq TRACMAX_{jk}, \quad j \in DN, k \in DN, j \neq k, t \in T \tag{9}$$

$$TRACMIN_{jk} \leq \sum_{i \in P} SQ_{ijkt}, \quad j \in DN, k \in DN, j \neq k, t \in T \tag{10}$$

Safety stock policy

Constraints (11) ensure that the inventory of each product i at each node j at each time period t must be higher than the required safety stock level for that product in that node.

$$SS_{ij} \leq FI_{ijt}, \quad i \in P, j \in DN, t \in T \tag{11}$$

Non-negativity and binary conditions

As defined above, the model uses both positive continuous variables (12) and binary variables (13).

$$SQ_{ijkt}, FI_{ijt}, LS_{ijt} \geq 0, \quad i \in P, j \in I, k \in I, t \in T \tag{12}$$

$$BV1_{ijt} \in \{0, 1\}, \quad i \in P, j \in I, t \in T \tag{13}$$

The above model is formed by constraints (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), and (13), using the objective function (1) that describes the proposed IM model. In order to compare the performance and adequacy of this model with classical inventory policies, two other models are developed: the continuous review inventory model (CR, Sect. 4) and the periodic review inventory model (PR, Sect. 5).

4 Continuous Review Inventory System Model (CR Model)

In the CR policy, the inventory level is continuously reviewed over the time horizon. Thus, whenever the inventory level is at or below a given reorder point level s, an order is placed that has a fixed quantity. Since inventory is tracked and the order is made when necessary, this inventory management policy is characterized by a fixed order quantity requested at variable time intervals. In order to represent this policy through a MILP model, two more parameters must be added to the list proposed in Sect. 3:

RSC_{ik} reference stock level of product i at node k (used for regional warehouses and retailers)
s_{ik} reorder point level of product i at node k (used for regional warehouses and retailers)

The fixed order quantity is then given by $(RSC_{ik} - s_{ik})$ since at the moment that an order is placed the inventory position (inventory level plus inventory in transit) must reach the reference stock level. In terms of model representation, the reference stock value can replace the BGM value used in constraints (6) and (7), since that value also works as an upper bound of any SQ variable. The MILP model that represents the continuous review policy consists of constrains (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), and (13), with the change in constraints (6) and (7), while the objective function (1) remains equal.

5 Periodic Review Inventory System Model (PR Model)

In this policy, the inventory level is reviewed at fixed time points, determined by a fixed review period. If at that time the inventory level is below a reference stock RS, an order is placed. The order quantity is determined by the difference between the reference stock and the current inventory level, to bring the inventory position up to level RS; otherwise, nothing is done until the next review point. A periodic inventory system uses variable order sizes at fixed time intervals. We add a subset of T that includes all the time periods for which there will be orders.

$ST \subset T$ all the time periods for which there will be orders due to the periodic review policy.

As in Sect. 4 also in here is necessary to add an extra parameter to the initial list of Sect. 3.

RSP_{ik} reference stock level of product i at node k (used for regional warehouses and retailers)

The original constraints (6) and (7) are now replaced by constraints (14) and (15).

$$SQ_{i0kt} + \sum_{l \in W \wedge l \neq k} SQ_{ilkt} \leq (RSP_{ik} - FI_{ikt}) \times BV1_{ikt}, \quad i \in P, k \in W, t \in ST \tag{14}$$

$$\sum_{j \in W} SQ_{ijkt} + \sum_{l \in R \wedge l \neq k} SQ_{ilkt} \leq (RSP_{ik} - FI_{ikt}) \times BV1_{ikt}, \quad i \in P, k \in R, t \in ST \tag{15}$$

In order to solve the non-linearity on the right end-side of constraints (14) and (15), we define the auxiliary variable Y_{ikt} (positive continuous variable) that represents the inventory of product i at node k at the end of time period t. Thus, the non-linear term $FI_{ikt} \times BV1_{ikt}$ is replaced by the continuous variable Y_{ikt} in constraints (14) and (15). The value of this auxiliary variable is given by Eq. (16):

$$Y_{ikt} = FI_{ikt} \times BV1_{ikt}, \quad i \in P, k \in I, t \in T \tag{16}$$

Using the definition of variable Y_{ikt} it is possible to impose the logical conditions (17) and (18). If the binary variable $BV1_{ikt}$ is 0, then the auxiliary variable Y_{ikt} is also 0 (condition (17)). If, on the other hand, the binary variable $BV1_{ikt}$ is equal to 1, we want to ensure that the new auxiliary variable takes the value of the inventory in the current time interval (FI_{ikt}), as expressed in condition (18).

$$BV1_{ikt} = 0 \implies Y_{ikt} = 0, \quad i \in P, k \in I, t \in T \tag{17}$$

$$BV1_{ikt} = 1 \implies Y_{ikt} = FI_{ikt}, \quad i \in P, k \in I, t \in T \tag{18}$$

To translate these logical conditions into the MILP model representation of the periodic review policy, we need to add the extra constraints (19), (20) and (21).

$$Y_{ikt} - RS_{ik} \times BV1_{ikt} \leq 0, \quad i \in P, k \in I, t \in T \tag{19}$$

$$-FI_{ikt} + Y_{ikt} \leq 0, \quad i \in P, k \in I, t \in T \tag{20}$$

$$FI_{ikt} - Y_{ikt} + RS_{ik} \times BV1_{ikt} \leq RS_{ik}, \quad i \in P, k \in I, t \in T \tag{21}$$

where RS_{ik} is the upper bound for FI_{ikt} (and hence also for Y_{ikt}). Constraints (19) and (20) ensure that the auxiliary variable takes the value of 0 if the binary variable is equal to 0 (constraints (19)). If this variable is equal to 1, then the auxiliary variable takes, at most, the value of FI_{ikt} (constraints (20)). In order to ensure that in this situation the auxiliary variable takes exactly the value of FI_{ikt}, we add

constraints (21). Note that this equation only becomes active whenever $BV1_{ikt} = 1$. Finally, we add constraints (22) and (23) that are applied for time periods where no orders are to be placed. They impose that no transportation activity occurs at these time periods.

$$SQ_{i0kt} + \sum_{l \in W \wedge l \neq k} SQ_{ilkt} = 0, \quad i \in P, k \in W, t \notin ST \tag{22}$$

$$\sum_{j \in W} SQ_{ijkt} + \sum_{l \in R \wedge l \neq k} SQ_{ilkt} = 0, \quad i \in P, k \in R, t \notin ST \tag{23}$$

The periodic review policy MILP model consists of constraints (2), (3), (4), and (5), (8),(9), (10), (11), (12), and (13), (16) and constraints (19), (20), (21), (22), and (23), using objective function (1).

6 Inventory Management Case Study

In this section we present a case study based on a Portuguese Company that we use to compare the three inventory management policies modeled, and to test the proposed IM policy. Due to confidentiality reasons the data provided has been changed but still describes the real operation. Please note that the products amounts involved in the present case study although referred as units they correspond to euro-pallets. The models were implemented in GAMS 23.5 modeling language and solved using the CPLEX 12.2.0.0 solver in an Intel CORE i5 CPU 2.27 GHz and 4 GB RAM. The stopping criterion was either a computational time limit of 3600 s or the determination of the optimal solution.

6.1 Data and Parameters

The supply chain considered involves one central warehouse, three regional warehouses and four retailers. Three families of products are considered, which for sake of simplicity will be modeled aggregated per family. The available storage capacity of all warehouses and retailers is of 500 SKU units and the transportation capacity is between 0 and 500 SKU units. We consider that demand at retailers is represented by a normal distribution, with parameters presented in Table 4. These values are given by the company forecast process. A 7-period planning horizon was assumed to test the three different inventory management policies. The first scenario considers the IM policy (modeled in Sect. 3), which uses a variable order quantity at variable time intervals. The second scenario explores the CR policy (Sect. 4) that uses a fixed order quantity at variable time intervals. Finally, the third scenario explores the PR policy (Sect. 5) where a variable order quantity at fixed time intervals is considered. Tables from 1 to 5 present all the parameters.

Table 1 Models parameters (euro)

Parameters	Values
OC for warehouses and retailers	20
HOC for warehouses	0.2
HOC for retailers	0.6
HTCW from central warehouse to regional warehouses	0.3
HTCR from regional warehouses to retailers	0.9
LS for retailers	25

Table 2 Unit transportation cost for all products (euro)

		Warehouse1	Warehouse2	Warehouse3
TRCW	Warehouse0	0.55	0.22	0.70
TSCW	Warehouse1	0	0.35	0.75
	Warehouse2	0.35	0	0.40
	Warehouse3	0.75	0.40	0

		Retailer1	Retailer2	Retailer3	Retailer4
TRCR	Warehouse1	0.22	0.20	0.32	0.38
	Warehouse2	0.68	0.52	0.34	0.10
	Warehouse3	0.95	0.70	0.40	0.25
TSCR	Retailer1	0	0.10	0.40	0.65
	Retailer2	0.10	0	0.15	0.50
	Retailer3	0.40	0.15	0	0.18
	Retailer4	0.65	0.50	0.18	0

Table 3 Initial inventory level (Ito)/safety stock (SS) on warehouses and retailers (unit)

		Warehouse1	Warehouse2	Warehouse3	Retailer1	Retailer2	Retailer3	Retailer4
Ito	Product1	45	30	40	24	22	20	18
	Product2	15	11	12	16	14	12	10
	Product3	11	9	10	8	4	6	9
		Warehouse1	Warehouse2	Warehouse3	Retailer1	Retailer2	Retailer3	Retailer4
SS	Product1	14	14	14	7	3	3	3
	Product2	11	11	11	2	2	2	2
	Product3	8	8	8	2	2	1	1

Table 4 Customer demand (CD) parameters for product1/product2/product3 (unit)

		Average demand	Standard deviation
CD	Retailer1	12 / 8 / 4	4 / 4 / 2
	Retailer2	11 / 7 / 4	4 / 4 / 3
	Retailer3	10 / 6 / 4	6 / 3 /1
	Retailer4	9 / 5 / 5	3 / 3 / 1

Table 5 Transportation lead time (LTT) for all products (time period)

		Warehouse 1	Warehouse 2	Warehouse 3	Retailer 1	Retailer 2	Retailer 3	Retailer 4
LTT	Warehouse0	2	1	3	NA	NA	NA	NA
	Warehouse1	NA	1	1	1	1	1	1
	Warehouse2	1	NA	1	2	2	1	0
	Warehouse3	1	1	NA	3	2	1	1
	Retailer1	NA	NA	NA	NA	1	1	1
	Retailer2	NA	NA	NA	1	NA	1	1
	Retailer3	NA	NA	NA	1	1	NA	1
	Retailer4	NA	NA	NA	1	1	1	NA

Table 6 Reference stock level for CR policy (RSC)/reference stock level for PR policy (RSP) on warehouses and retailers (unit)

		Warehouse 1	Warehouse 2	Warehouse 3	Retailer 1	Retailer 2	Retailer 3	Retailer 4
RSC /RSP	Product1	280	280	280	72	66	60	54
	Product2	220	220	220	48	42	36	30
	Product3	160	160	160	36	30	28	20

Table 6 illustrates the reference stock level for the CR and PR policies for the warehouses and retailers. The reorder point value is assumed equal to the safety stock. In the PR policy that implies a review period taking place every three time periods, the first time period with revision is time period 3. The initial inventory level, the customer demand, reference stock level for warehouses and retailers and the supply chain structure are the same for the three policies.

6.2 Experimental Results

The costs per nature of the objective function analyses for all three policies are present below, although computational statistics are shown in appendix.

6.2.1 Initial Inventory as a Parameter Given by the Portuguese Company

The costs per nature of the objective function for all three inventory policies are shown in Table 7. For the IM policy the holding cost, of 557.60 euro, is the highest cost term. The objective function reaches a value of 1955.46 euro. For the CR policy the objective function has a higher cost value than the observed for the IM policy, being the cost with highest contribution the lost sales cost. Holding costs also

Table 7 Costs per nature for all three inventory policies (euro)

	IM policy		CR policy		PR policy	
Inventory policy	Value	Percentage	Value	Percentage	Value	Percentage
Ordering	520.00	26.59	380.00	10.26	160.00	1.56
Holding	557.60	28.52	1026.00	27.71	587.80	5.71
Holding in transit	168.00	8.59	559.50	15.11	273.60	2.66
Transportation	135.48	6.93	286.58	7.74	121.02	1.18
Transshipping	124.38	6.36	100.91	2.73	18.81	0.18
Lost-sales	450.00	23.01	1350.00	36.45	9125.00	88.71
Total	1955.46	100.00	3702.99	100.00	10,286.23	100.00

Table 8 Costs per nature for CR policy for three different reference stock levels (euro)

	Minus 10 %		Equal to Table 7		Plus 10 %	
Stock level	Value	Percentage	Value	Percentage	Value	Percentage
Ordering	380.00	10.80	380.00	10.26	380.00	9.78
Holding	913.56	25.96	1026.00	27.71	1116.72	28.75
Holding in transit	528.18	15.01	559.50	15.11	606.78	15.62
Transportation	265.28	7.54	286.58	7.74	312.40	8.05
Transshipping	81.70	2.32	100.91	2.73	118.62	3.05
Lost-sales	1350.00	38.37	1350.00	36.45	1350.00	34.75
Total	3518.72	100.00	3702.99	100.00	3884.52	100.00

increase when compared to the IM policy. The PR policy has the lowest ordering costs. On the other hand, since inventory levels are not replenished frequently, the highest cost becomes associated with the lost sales. The objective function value is 10,286.23 euro, being the highest one amongst the three policies. In conclusion the IM policy appears as more flexible than the CR or PR policies, which is then confirmed by the lower operational costs. This is explained by the policy flexibility in terms of managing the flows allowing for an order occurrence strictly when necessary. This leads to a less costly system operation.

The costs per nature for CR policy for three different reference stock levels are shown in Table 8. The ordering and lost-sales costs are the same for all situations, while that the remaining costs increase with the reference stock level increase. Note that the scenario with less 10 % yielded better solutions without increasing the lost sales, therefore as a recommendation, the company should reduce the reference stock level in 10 %.

Table 9 shows the costs per nature for PR policy for three different review periods. When the review period length increases, we have in general a decrease of the costs per nature except lost-sales. This is due to the fact that we do not have enough stock to satisfy the customers during the review period.

Table 9 Costs per nature for PR policy for three different review periods (euro)

	1 time period		Equal to Table 7		5 time period	
Review period	Value	Percentage	Value	Percentage	Value	Percentage
Ordering	420.00	11.66	160.00	1.56	20.00	0.18
Holding	949.80	26.37	587.80	5.71	412.80	3.67
Holding in transit	522.00	14.49	273.60	2.66	51.30	0.46
Transportation	269.06	7.46	121.02	1.18	19.36	0.17
Transshipping	91.32	2.54	18.81	0.18	0.00	0.00
Lost-sales	1350.00	37.48	9125.00	88.71	10,750.00	95.52
Total	3602.18	100.00	10,286.23	100.00	11,253.46	100.00

Table 10 Costs per nature for all three inventory policies (euro)

	IM policy		CR policy		PR policy	
Inventory policy	Value	Percentage	Value	Percentage	Value	Percentage
Ordering	320.00	31.19	140.00	11.93	20.00	1.64
Holding	581.80	56.70	982.80	83.72	1191.40	97.96
Holding in transit	0.00	0.00	0.90	0.07	0.00	0.00
Transportation	1.40	0.14	5.72	0.49	2.20	0.18
Transshipping	122.77	11.97	44.55	3.79	2.70	0.22
Lost-sales	0.00	0.00	0.00	0.00	0.00	0.00
Total	1025.97	100.00	1173.97	100.00	1216.30	100.00

6.2.2 Initial Inventory as an Optimization Variable

Now we run the same analyses, but considering that the initial inventory is subject to optimization, in order to analyze the possibility of having zero lost-sales and consequently maximum service level in all retailers. The costs per nature of the objective function for all three inventory policies are shown in Table 10. As expected the most significant value is the holding cost to face demand in the initial time periods, in order to avoid lost-sales.

The costs per nature for CR policy for three different reference stock levels are shown in Table 11. The holding costs are the most representative for all situations. This results confirm that the company should reduce the reference stock level in 10 %.

Table 12 shows the costs per nature for PR policy for three different review periods. With the review period increase, we have a generally decrease of the costs per nature except holding cost. This is due to the fact that we must have enough products to satisfy the customers during the review period.

Table 11 Costs per nature for CR policy for three different reference stock levels (euro)

	Minus 10 %		Equal to Table 10		Plus 10 %	
Stock level	Value	Percentage	Value	Percentage	Value	Percentage
Ordering	140.00	12.00	140.00	11.93	100.00	8.49
Holding	969.12	83.07	982.80	83.72	1032.36	87.62
Holding in transit	1.44	0.12	0.90	0.07	0.00	0.00
Transportation	5.37	0.46	5.72	0.49	0.00	0.00
Transshipping	50.64	4.35	44.55	3.79	45.87	3.89
Lost-sales	0.00	0.00	0.00	0.00	0.00	0.00
Total	1166.57	100.00	1173.97	100.00	1178.23	100.00

Table 12 Costs per nature for PR policy for three different review periods (euro)

	1 time period		Equal to Table 10		5 time period	
Review period	Value	Percentage	Value	Percentage	Value	Percentage
Ordering	220.00	18.46	20.00	1.64	20.00	1.64
Holding	896.00	75.18	1191.40	97.96	1194.20	98.13
Holding in transit	0.00	0.00	0.00	0.00	0.00	0.00
Transportation	5.20	0.44	2.20	0.18	2.80	0.23
Transshipping	70.54	5.92	2.70	0.22	0.00	0.00
Lost-sales	0.00	0.00	0.00	0.00	0.00	0.00
Total	1191.74	100.00	1216.30	100.00	1217.00	100.00

7 Conclusions

This paper proposes a generic inventory management policy applied to a multi-period and multi-product supply chain, which provides an inventory and distribution plan that minimizes the total operation costs. Based on the experimental results, it could be concluded that the proposed IM policy proved to be more flexible since the total costs are lower when compared to the classical policies for the same network and under the same conditions.

The classical policies are very commonly used in companies that control more than one echelon even though they were designed to be applied to a single entity. These policies are well spread in most ERPs and commonly used by decision makers. Given this fact, it is important to show why incorporating all the costs in the optimal policy is important as well as when capacities are disputed by more than one product.

In the real world there are thousands of SKUs, but simplifying techniques are used, such as grouping in specific families. Nevertheless, a tighter formulation should be deigned in future.

The lost sales represent a large value that reflects somewhat the impact of those lost sales for our test company, thus other model driving force should be used,

namely profit or even to balance costs with service level. Our future work also will focus on optimizing the safety stock as well to reduce lost sales.

The results obtained and consequent conclusions are related with the test company. In future works, more instances should be studied so as to show the potential generalization of our conclusions.

Appendix: Experimental Results Regarding to Computational Statistics

In this Appendix, we show the computational statistics of the inventory management case study.

A1. Initial Inventory as a Parameter Given by the Portuguese Company

The computational statistics for all three inventory policies are shown in Table 13. Regarding to computational time used, the IM model has the highest one, since all decisions are taken by the optimization model.

Computational statistics for CR policy for three different reference stock levels are shown in Table 14. Regarding to the computational time used, its increases with the reference stock level increase, but the required equations and variables maintain the same value.

Table 13 Computational statistics for all three inventory policies (0 % gap)

Inventory policy	IM policy	CR policy	PR policy
MIP solution	1955.46	3702.99	10,286.23
Single equations	1112	1112	1238
Single variables	1378	1378	1420
Discrete variables	168	168	168
Computational time (second)	265.26	4.43	0.44

Table 14 Computational statistics for CR policy for three different reference stock levels (0 % gap)

Stock level	Minus 10 %	Equal to Table 13	Plus 10 %
MIP solution	3518.72	3702.99	3884.52
Single equations	1112	1112	1112
Single variables	1378	1378	1378
Discrete variables	168	168	168
Computational time (second)	3.18	4.43	6.02

Table 15 Computational statistics for PR policy for three different review periods (0 % gap)

Review period	1 time period	Equal to Table 13	5 time period
MIP solution	3602.18	10,286.23	11,253.46
Single equations	1553	1238	1175
Single variables	1525	1420	1399
Discrete variables	168	168	168
Computational time (second)	5.41	0.44	0.17

Table 16 Computational statistics for all three inventory policies

Inventory policy	IM policy	CR policy	PR policy
MIP solution	1025.97	1173.97	1216.30
Best possible	1012.80	1173.97	1216.30
Relative gap	1.28 %	0 %	0 %
Single equations	1112	1112	1238
Single variables	1399	1399	1441
Discrete variables	168	168	168
Computational time (second)	3600	2.92	0.06

Computational statistics for PR policy for three different review periods is present in Table 15. The results show that the number of equations and variables decreases with the increase of the review period while computational time used decreases. This behavior is expected, since with less revisions, less constraints and variables are required.

A2. Initial Inventory as an Optimization Variable

Computational statistics for all three inventory policies is present in Table 16. For IM policy we have 1.28 % of relative gap after one hour of computation, but we already reached the lower total costs among all policies. The complexity of the IM model is expected, since all decisions are taken by the optimization model.

Table 17 shows the computational statistics for CR policy for three different reference stock levels. The computational time used increase with the reference stock increase, but the equations and variables maintain the same value.

Computational statistics for PR policy for three different period review in Table 18, shows that single equations and variables decrease with the period review increase, while computational time used decrease.

Table 17 Computational statistics for CR policy for three different reference stock levels (0 % gap)

Stock level	Minus 10 %	Equal to Table 16	Plus 10 %
MIP solution	1166.57	1173.97	1178.23
Single equations	1112	1112	1112
Single variables	1399	1399	1399
Discrete variables	168	168	168
Computational time (second)	1.42	2.92	4.48

Table 18 Computational statistics for PR policy for three different review periods (0 % gap)

Review period	1 time period	Equal to Table 16	5 time period
MIP solution	1191.74	1216.30	1217.00
Single equations	1553	1238	1175
Single variables	1546	1441	1420
Discrete variables	168	168	168
Computational time (second)	4.11	0.06	0.06

References

1. Abdul-Jalbar, B., Gutiérrez, J., Sicilia, J.: Single cycle policies for the one-warehouse N-retailer inventory/distribution system. OMEGA Int. J. Manag. Sci. **34**, 196–208 (2006)
2. Axsater, S.: Approximate optimization of a two-level distribution inventory system. Int. J. Prod. Econ. **81**, 545–553 (2003)
3. Beamon, B.M.: Supply chain design and analysis: models and methods. Int. J. Prod. Econ. **55**, 281–294 (1998)
4. Giannoccaro, I., Pontrandolfo, P.: Inventory management in supply chains: a reinforcement learning approach. Int. J. Prod. Econ. **78**, 153–161 (2002)
5. Hsiao, Y.-C.: Optimal single-cycle policies for the one-warehouse multi-retailer inventory/distribution system. Int. J. Prod. Econ. **114**, 219–229 (2008)
6. Krautter, J.: Inventory theory: new perspectives for cooperative management. Int. J. Prod. Econ. **59**, 129–134 (1999)
7. Monthatipkul, C., Yenradee, P.: Inventory/distribution control system in a one-warehouse/multi-retailer supply chain. Int. J. Prod. Econ. **114**, 119–133 (2008)
8. Ozdemir, D., Yucesan, E., Herer, Y.: Multi-location transshipment problem with capacitated transportation. Eur. J. Oper. Res. **175**, 602–621 (2006)
9. Paterson, C., Kiesmuller, G., Teunter, R., Glazebrook, K.: Inventory models with lateral transshipments: a review. Eur. J. Oper. Res. **210**, 125–136 (2011)
10. Yousuk, R., Luong, H.T.: Modeling a two-retailer inventory system with preventive lateral transshipment using expected path approach. Eur. J. Ind. Eng. **7**, 248–274 (2013)

Periodic Versus Non-periodic Multipurpose Batch Plant Scheduling: A Paint Industry Case Study

Miguel Vieira, Tânia Pinto-Varela, and Ana Paula Barbosa-Póvoa

Abstract In order to guarantee the correct plant resources allocation, an efficient and uniform methodology is required to address the wide diversity of operational problems. The high flexibility requirement in the process industry, to be able to satisfy the market demand and service level, triggered the development of optimal scheduling methodologies in industrial management. In this work, a case study of a chemical process in the paint industry is presented, where the complexity of resource allocation and schedule optimisation is addressed through the use of the Resource Task Network (RTN) methodology. Two scheduling models are compared, considering a non-periodic and periodic operation mode. Mixed Integer Linear Programming (MILP) formulation are implemented where profit is maximized. The results of each formulation applied to a real case study are compared and discussed based in plant schedule resources allocation and required manpower.

1 Introduction

Global competition and fast changing economic conditions have been shaping industrial processes by requiring the multiple coordination of products, recipes and equipment. For that purpose, the development of computer-based decision-support tools to assist the optimal scheduling of operations and resources is becoming decisive for improving process performance in plants with high process flexibility. Furthermore, current facilities are also envisaging plant redesign to adjust multiple production requirements to a competitive market.

Reviewed literature demonstrates the extensive research interest in this area. With focus in scheduling problems, the challenge has been addressed not only from the mathematical point of view, but also its relevance in solving real case studies [6]. The development of a scheduling system requires a model formulation grounded in an adequate methodology to characterize the problem without ambiguities.

M. Vieira (✉) • T. Pinto-Varela • A.P. Barbosa-Póvoa
CEG-IST, University of Lisbon
e-mail: migueljvieira@tecnico.ulisboa.pt; tania.pinto.varela@tecnico.ulisboa.pt; apovoa@tecnico.ulisboa.pt

J.P. Almeida et al. (eds.), *Operational Research*, CIM Series in Mathematical Sciences 4, DOI 10.1007/978-3-319-20328-7_24

The problem characterization can be done using two main methodologies: one states that design and scheduling problems are very diverse and require specific models; the other assumes similarity between the problems and proposes a uniform representation. The latter was explored by Kondili et al. [5] and Shah et al. [12], proposing the first generic representation for scheduling batch processes called State-Task-Network (STN). Later on, Pantelides [8] suggested the Resource-Task Network (RTN), a novel uniform methodology able to address the optimal allocation of resources to tasks. Both STN and RTN representations have been successfully used for modelling scheduling problems: Castro et al. [2] developed a continuous RTN formulation based on a uniform time grid for the scheduling of multipurpose batch plants; Pinto et al. [10] addressed a comparative set of examples to evaluate the adequacy and effectiveness of STN/m-STN/RTN formulations to the design of multipurpose batch plants; later, the same authors, Pinto et al. [11], proposed the detailed design and schedule of batch plants to address the problem of uncertainty associated with production demand, considering a cyclic (periodic) operation; Chen and Chang [3] considers the periodic scheduling of multipurpose batch plants based in a RTN continuous representation, with the integration of heat recovery problems; Moniz et al. [7] proposed a RTN discrete-time sequential approach for the simultaneous scheduling of regular and non-regular products in multipurpose plants, applied to a real scheduling problem from the chemical-pharmaceutical industry; and Shaik and Vooradi [13] proposed a unified framework and developed two unit-specific event-based approaches for STN and RTN representations.

The scientific research reveals the widespread of mathematical modelling applied to the scheduling of complex networks, exploring diverse operational problems characteristics (e.g. variable/fixed batch size, storage/transfer policies, energy integration, changeovers, discrete/continuous time representation) or multiple and different objective functions (e.g. makespan, earliness, or cost minimization). Relevant reviews of state-of-the-art optimisation methods for short-term scheduling and design of batch processes can be found, such as: Méndez et al. [6] classified the scheduling problems according to its process topology, detailing the different modelling and optimisation techniques, with focus on both discrete and continuous time models; Barbosa-Póvoa [1] analysed both the grassroots and the retrofit design problems on multi-product and multipurpose batch plants, linked to scheduling problems to address production flexibility; and Verderame et al. [15] provided an overview over a number of independent sectors exploring different programming methodologies to address uncertainty within the planning and scheduling problems. The mathematical programming is the most common approach for multiproduct/multipurpose plant cases, where both Mixed Integer Linear and Non-Linear Programming (MILP and MINLP) formulations are considered. However, the increasing model complexity have endured the discussion over alternative mathematical formulations, exploring decomposition approaches such as problem oriented heuristics, evolutionary algorithms and meta-heuristics (Simulated Annealing, Tabu Search and Ant Colonies) [4, 14].

This work considers a real case study of a paint company, addressing the problem complexity of resource allocation with the development of an optimisation mathe-

matical model. The aim is to determine the optimal scheduling while considering the profit maximization, analysing the required manpower to execute the demanded production plan with different operation modes (non-periodic and periodic).

The remainder of the paper is structured as follows: Sect. 2 presents the problem characterization, followed by the modelling framework detailing the proposed formulations in Sect. 3; Sect. 4 present the case study results and the paper finalizes with the main conclusions and future remarks.

2 Problem Statement

Process scheduling includes the assignment of tasks, in a specific sequence, to the available resources during a time line. This assumes special complexity when competition among the scheduled tasks for limited available resources is verified. The characterization of the process allows the identification of different constraints that need to be considered in the mathematical model, such as: availability and capacities of equipment, quantities of material resources available, manufacturing of different products using different recipes or/and tasks' processing times, demand and service levels, among others.

The case study encompasses a paint manufacturing process addressing a scheduling problem: the process management requires an optimal sequence of operations, considering all the aspects of the manufacturing process, while maximizing resource efficiency and satisfying demand levels. Since the labour costs are one of the most significant in the operational structure, its resources optimisation represents one of the most important aspect in the plant management. With the requirement of high flexibility in the production output, the access to decision-support tools is essential to provide information concerning the required manpower. The company also wants to study the ability to perform a single campaign operating in a periodic mode, comparing the results with a non-periodic scheduling problem (single production plan). These aspects are now engaged into two proposed models: model M1 follows the extension of the formulation presented by Vieira et al. [16], with the resources allocation performed simultaneously with the optimal scheduling characterization, using a non-periodic of operation; and model M2 presents the extension of the previous model, exploring a periodic operation mode.

The scheduling problem can be stated as follows, assuming a uniform discretization of time:

Given:

- The product recipes in terms of their respective RTN framework (tasks and resources required);
- Resources availability (raw materials and equipment) and task suitability;
- Time horizon;
- Set of final products composing a demand plan for the given time horizon;
- Characteristics of the processing units and processing time of each task;
- Operational costs and value of products.

Determine:

- Process schedule;
- Optimal resources allocation to each task;
- Manpower characterization for the production plan.

So as to maximize the profit for the time horizon.

3 Mathematical Formulation

The proposed mathematical formulation is based on the representation proposed by Pantelides [8], the Resource-Task Network, a uniform concept methodology that involves two types of entities, Tasks and Resources. A Task is an abstract operation that consumes and/or produces a specific set of indistinctive Resources. Resources can be classified as non-renewable, such as raw materials or utilities, and renewable, which considers the remaining resources on the plant (equipment or manpower). In this paper, it is assumed that all equipment resources, with exception of storage tanks, are considered individually. Thus, if two or more identical pieces of equipment exists, one resource will need to be defined for each item. Furthermore, it is assumed that only one task can be executed in any given equipment resource at a certain time.

The RTN methodology was applied in a preliminary formulation proposed by Vieira et al. [16]. In that previous work, the optimal schedule was determined for two simple problem cases with distinct sets of available manpower, for the profit maximization. The model is now extended to integrate the manpower as a variable, allowing the optimisation of several resources usage such as raw materials, equipment units and manpower. The following indices, sets, parameters and variables are defined.

Indices
k – tasks
r – resource (processing unit, intermediary or final product)
t – time

Sets
E – processing units
F – final products
I – intermediate storage materials

(continued)

K_r – tasks that require resource r
M – material resources
P – products
R – production resources

Parameters

$\mu_{kr\theta}$ – allocation/release coefficient of resource r (processing unit and manpower) in task k at time θ relative to the start of the task

τ_k – processing time of task k

$v_{kr\theta}$ – production/consumption proportion of resource (intermediary or final product) r in task k at time θ relative to the start of task

c_r – cost of manufacturing products r

c_{HR} – operational cost of each element of resource HR

H – scheduling horizon

R_{rt}^{max} – maximum resource availability of resource r (intermediary or final product) at time interval t

R_{r0} – resource r availability in the beginning of the planning horizon

T – cycle time

v_r – value of product r

V_{kr}^{min}, V_{kr}^{max} – minimum and maximum capacity of resource r (processing unit) for task k

Variables

ξ_{kt} – batch size of task k at time interval t

$\Omega(t)$ – wrap-around time operator

N_{kt} – binary variables that are equal to 1 if task k starts at time interval t

$R_{HR,0}$ – allocation of resource HR at the beginning of the scheduling horizon

R_{rt} – resource availability r at time interval t

The framework supports the development of a MILP model, using a discrete time representation, considering three main constraints: excess resource balance, excess resource capacity and operational constraints. Each task k has a fixed duration τ_k and its execution, starting at time t, is characterized by its extent – a pair of variables (N_{kt}, ξ_{kt}). N_{kt} is an integer variable representing the start of task k at event point t, while ξ_{kt} is a continuous variable defining the amount of resource processed by task

k at event point t. The amount of resource r consumed or produced at each time $\theta = 0, \ldots, \tau_k$ is expressed by $(\mu_{kr\theta} N_{k,t-\theta} + \nu_{kr\theta} \xi_{k,t-\theta})$. The coefficients $\mu_{kr\theta}$ and $\nu_{kr\theta}$ represent the discrete interaction linked to the two variables considering the resource r consumption/production in task k and time θ. Negative values for the latter indicate consumption of resource, while positive values denote production. Changes to the resource utilization can only occur at interval boundaries. The variable R_{rt} keeps track of the resource availability r at time t and the change in the excess resource level for each resource type, from one time interval to the next, is given by excess resource balance constraint.

3.1 Non-periodic Operation Mode Formulation: M1

The following model constraints (M1) consider the manpower characterization and profit maximization for the time production horizon in a non-periodic operation.

Excess resource balance: the mass balance for each resource r must be satisfied at every instant t, considering R_{r0} the level of resource available initially (K_r is the set of tasks that use resource r of R production of resources and H the planning horizon).

$$R_{rt} = R_{r,t-1} + \sum_{k \in K_r} \sum_{\theta=0}^{\tau_k} (\mu_{kr\theta} N_{k,t-\theta} + \nu_{kr\theta} \xi_{k,t-\theta}) \quad \forall r \in R, t \in 1, \ldots, H \tag{1}$$

With the profit maximization, the formulation addresses the determination of the optimal (minimum) manpower by considering the resource *HR* as a multiple resource, allowing the initial availability $R_{HR,0}$ to be greater than 1. In this case, $R_{HR,0}$ is defined as a discrete variable that represents the team of operators composing the resource *HR* for the initial point of the time horizon.

Excess Resource capacity constraint: the amount of excess resource r must be, during the time planning horizon, a positive amount and lower or equal than the maximum production plan demand.

$$0 \leq R_{rt} \leq R_{rt}^{max} \quad \forall r \in R, t \in H \tag{2}$$

Operational constraints: for each task k taking use of a processing equipment resource r, the amount of material being processed must always be within the maximum and minimum capacities available, given by V_{kr}, where E is the subset of R for the processing units.

$$V_{kr}^{min} N_{kt} \leq \xi_{kt} \leq V_{kr}^{max} N_{kt} \quad \forall r \in E \subset R, k \in K_r, t \in H \tag{3}$$

The objective function considers the profit maximization, adding the associated production data of each resource r where F is a subset r of the final products R. v_r

and c_r represent the value and production cost of each final product, and c_{HR} the cost of each human resource *HR*.

$$max\left[\left(\sum_{t=H} R_{rt}(v_r - c_r)\right) - (R_{HR,0} \times c_{HR})\right] \quad \forall r \in F \subset R \tag{4}$$

3.2 Periodic Operation Mode Formulation: M2

Based on the formulation M1 and considering a more adequate case study characterization, the results of the implementation of a periodic operation mode are analysed. In a periodic operation the concept of cycle time T is used as the shortest time interval at which a cycle is repeated. A cycle represents a sequence of operations involving the production of all desired products and utilization of all available resources. Since all cycles are equal, the problem is formulated as a non-periodic operation over a single cycle where it is guaranteed that the process at the beginning and at the end of the cycle is the same. This idea was developed by Shah et al. [12], considering that the execution of a task is allowed to overlap successive cycles, and since the cycle is repeated, its execution is modelled by wrapping around to the beginning of the same cycle. The wrap-around operator is given by (5).

$$\Omega(t) = \begin{cases} t, & \text{if } t \geq 1 \\ \Omega(t+T), & \text{if } t \leq 0 \end{cases} \tag{5}$$

When this is applied, for instance, to the variable $N_{k,\Omega(t-\theta)}$, for $t-\theta \leq 0$ leads to a resource allocation that will start as time $(t-\theta+T)$. For the time discretization, the planning horizon H is divided into n equal cycles of duration T (Fig. 1). Each cycle is then divided into a number of elementary steps of fixed duration δ, beginning at $t = 1$ and ending at $t = T + 1$, where the latter coincides with the starting point of the next cycle [9].

For the excess resource balance, different types of constraints were considered concerning the types of resources involved. The renewable resource balances (equipment resources subset *E* and *HR*) at each time t are related with the amount of equipment in the previous instant and the amount produced or consumed in instant t.

$$R_{rt} = R_{r,\Omega(t-1)} + \sum_{k \in K_r} \sum_{\theta=0}^{\tau_k} (\mu_{kr\theta} N_{k,\Omega(t-\theta)}) \quad \forall r \in E \cup HR, t = 1, \ldots, T \tag{6}$$

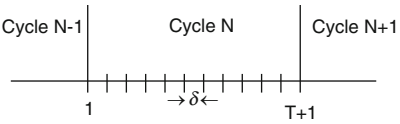


Fig. 1 Time discretization for a single cycle [9]

For the intermediate storage materials and HR resources it is necessary to guarantee that the initial quantity at the beginning and at the end of the cycle is the same, expressed in Eq. 7.

$$R_{rt|t=1} = R_{rt|t=T+1} \qquad \forall r \in I \cup HR \tag{7}$$

For the non-renewable resource balances (materials resources subset M) the same balance principle is followed, where R_{r0} is the amount available initially (e.g. raw material).

$$R_{rt} = R_{r0|t=1} R_{r,\Omega(t-1)|t\geq 2} + \sum_{k\in K_r} \sum_{\theta=0}^{\tau_k} (v_{kr\theta}\xi_{k,\Omega(t-\theta)}) \qquad \forall r \in M \neq F \cup HR, t = 1,\ldots,T \tag{8}$$

For the final products F, the balance is given by Eq. 9.

$$R_{rt} = R_{r,\Omega(t-1)|t<t\leq T+1} + \sum_{k\in K_r} \sum_{\theta=0}^{\tau_k} (v_{kr\theta}\xi_{k,\Omega(t-\theta)|t<T+1}) \qquad \forall r \in F, t = 1,\ldots,T+1 \tag{9}$$

Thus, the periodic operation mode formulation, whose goal is to maximize the profit, can be briefly posed as follows:

Model M2

$$max \left[\left(\sum_{t=H} R_{rt}(v_r - c_r) \right) - (R_{HR,0} \times c_{HR}) \right] \qquad \forall r \in F \subset R \tag{4}$$

s.t. Constraints [3–9]

4 Case Study

Our case study is based on a real chemical industry process: the company is mainly dedicated to the development and production of water-based paint products, providing different compositions concerning its final application. The paint is composed by a binder, which is the film forming component; pigments, that give the colour; and solvent, which keeps the paint in the liquid form. The process is laid out vertically composed by three stages: an initial step of mixing of the raw materials (dispersion tanks), followed by a finishing step of homogenization of paint properties with temporary storage (finishing tanks) and ending in one of the filling machines (filling lines), where the paint is filled into package containers and sent to storage. The company has installed a large number of equipment units, suitable for the current demand, where some are dedicated to the production of a single paint category: Textured, Smooth or Brand type.

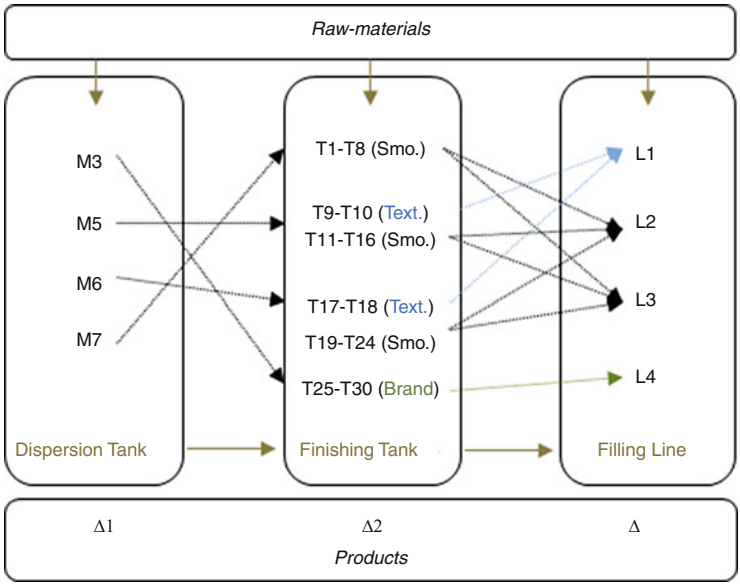


Fig. 2 Production process layout

In Fig. 2 are detailed the possible connections between the set of available equipment resources considered for our study: 4 dispersion tanks $\{M_3, M_5, M_6, M_7\}$, 30 finishing/storage tanks $\{T_1, \ldots, T_{30}\}$ and 4 filling machine lines $\{L_1, L_2, L_3, L_4\}$, corresponding to the most relevant equipment in this process.

The different connections between equipment units presents the possible paths of the process flow. For example, M_5 can produce two categories of paint products (*Textured* and *Smooth*) that can be stored in one of the allocated tanks – $\{T_9, T_{10}\}$ for Textured and $\{T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}\}$ for Smooth products – which then can be filled respectively in line L_1 for *Textured* and lines L_2 and L_3 for *Smooth* products.

Additional considerations in the characterization of the problem must also be considered:

- A production order has to be accomplished in 48 h, with the filling stage taking place in the following shift after the manufacturing step (dispersion and finishing), due to intermediate quality tests procedures. During this elapsed time, the batch is kept stored in the finishing tank.
- The manufacturing step presents a constant processing time, while the filling step verifies a higher variability in the processing of each order, depending on the batch size and number/types of containers to produce (*Small*, *Medium* or *Large* containers).
- To guarantee a steady process flow, the management team has decided to restrain the available manpower in manufacturing areas per shift, requesting additional temporary manpower only when production demand increases. A

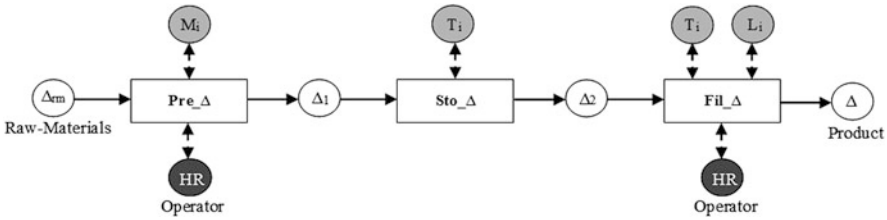


Fig. 3 RTN-based single-product process representation

team of operators is assigned to perform the process tasks, distributed to the manufacturing and filling operations. For this case study, it is assumed that all elements of the team are qualified to accomplish all tasks.

The objective is to generate the optimal schedule for a weekly production plan (five consecutive shifts of 8 h) maximizing the total profit, while guaranteeing the optimal allocation of the resources – equipment, material and manpower – considering a non-periodic and periodic (single campaign) operation mode.

To illustrate our case study and guarantee confidentiality requirements, the process recipe for generic product Δ is shown in Fig. 3, considering the RTN framework:

- The raw-material, Δ_{rm}, is used in the *Preparation* stage, *Pre_*Δ, producing the material, Δ_1, and sent it to storage; this *Storage* task, *Sto_*Δ, consumes Δ_1 and produces Δ_2, which goes to the *Filling* stage, *Fil_*Δ, until the total quantity of product Δ order is completed. The variables – Δ_1 and Δ_2 – are used to represent the connections between the tasks, zero-wait storage policy.
- Concerning Resources allocation during processing times:
  - The *Preparation* stage requires one equipment (M_i equipment, with $i = 3, 5, 6, 7$) and one team member *HR*;
  - The *Storage* stage requires one of the Tanks (T_i);
  - The *Filling* stage requires one of the lines (L_i filling lines, with $i = 1, 2, 3, 4$) and one team member *HR*. This task also captives the respective Tank T_i used in the previous *Storage* task, since the filling operation is completed only when the tank is emptied.
- For the Tanks allocation, three variables T_T, T_S and T_F were used, defining the dedicated groups of tanks, respectively, for *Textured*, *Smooth* and *Brand* paint types. These variables T_T, T_S and T_F will consider, groups of 4, 6 and 20 tanks, respectively, according to Fig. 2.

The three process stages are characterized as follows:

- *Preparation* stage – *Pre_*Δ: represents all the manufacturing operations (raw-materials dispersion and recipe homogenization) with an average processing time of 150 min, independent of the type of product.

- *Storage* stage – *Sto_Δ*: assuming a 48 h production time, it was settled an average time for the storage task of 330 min, during which the quality control and/or properties corrections should take place.
- *Filling* stage – *Fil_Δ*: the processing time of the Filling orders is dependent on the selected filling line, type of product, batch volume and type of containers format.

Regarding discretization time, a prior analysis to the processing rates has considered 30 min slots as a suitable time representation. Sales prices and production costs are identical for each product so not to bias the production of any particular product: $(v_r - c_r) = 100$ monetary units and $c_H R = 1000$ monetary units (m.u.).

4.1 Company Case Study

A standard production demand plan proposed by the company is shown in Table 1. Wherein a production of 28 orders for a total of 196 kL is solicited by the planning department to be accomplished in five consecutive 8 h shifts horizon (weekly plan), considering not only different types of paint (*Textured*, *Smooth* or *Brand*) but also different orders of containers format (*Large*, *Medium* or *Small* size), according to stock requirements.

As an example, product *G* demand is characterized as follows:

Order supply

- It is a *Smooth* paint type and has a demand for two orders of 9 kL, with 4 kL produced in *Large*, 3 kL in *Medium* and 2 kL in *Small* containers (at the filling stage);

Resource allocation and processing time

- It will be manufactured in machine M_7 using one of the team members *HR*, stored in one of T_F tanks and filled in L_3 by a *HR* team member;
- The processing times of each task are 150 min (M_7), 330 min (T_F) and 240 min (L_3).

According to the detailed information in Table 1, Table 2 summarizes the information highlighting the resources multipurpose characteristics, demonstrating the competition among product orders and between the equipment required for each task.

The production plan shown was executed in the facility using six operators available as manpower. Despite the 196 kL weekly production plan demanded, only 161 kL was fulfilled, representing 82 % of the initial plan.

Based in the previous data, our study addresses a generic formulation for the scheduling optimisation of a paint production process, with the aim of not only maximize the profit, but also maximize the resource utilization. To do that, the two model formulations already presented – M_1 and M_2 – are explored, where the

Table 1 Production plan data with resource allocation and associate processing times

							Resources/Proc.time(min)		
Production plan			Order(kL)	Fill.order/size(kL)			[HR/task]		[HR/task]
Product	Type	Mult.		L	M	S	Machine	Tank	Fill.line
A	Text.		10	8	2		M_5/150	T_T/330	L_1/390
B	Text.	2×	6	6			M_6/150	T_T/330	L_1/210
C	Text.		8	8			M_5/150	T_T/330	L_1/270
D	Smo.		6	4	2		M_5/150	T_F/330	L_2/120
E	Smo.		9	8		1	M_6/150	T_F/330	L_2/180
F	Smo.		6	3,5	2	0,5	M_5/150	T_F/330	L_3/120
G	Smo.	2×	9	4	3	2	M_7/150	T_F/330	L_3/240
H	Smo.		7	7			M_6/150	T_F/330	L_2/90
I	Smo.	2×	6	4	1	1	M_6/150	T_F/330	L_2/150
J	Smo.	3×	8	7	1		M_7/150	T_F/330	L_3/120
K	Smo.		5		5		M_7/150	T_F/330	L_3/90
Y	Smo.		6		5	1	M_7/150	T_F/330	L_3/120
M	Smo.	3×	10	8	2		M_7/150	T_F/330	L_3/120
N	Smo.		8	7	1		M_6/150	T_F/330	L_2/120
O	Smo.		5	4	1		M_7/150	T_F/330	L_3/90
P	Brand	4×	5	5			M_3/150	T_S/330	L_4/120
Q	Brand		4	4			M_3/150	T_S/330	L_4/90
R	Brand		6	6			M_3/150	T_S/330	L_4/120
TOTAL (28 ORDERS)			196						

Table 2 Production plan multipurpose characteristics

Equipment	Products	Paint types
M_3	P, Q, R	*Brand*
M_5	A, C, D, F, M	*Textured*/*Smooth*
M_6	B, E, H, I, N	*Textured*/*Smooth*
M_7	G, J, K, Y, O	*Smooth*
T_T	A, B, C	*Textured*
T_F	D, E, F, G, H, I, J, K, Y, M, N, O	*Smooth*
T_S	P, Q, R	*Brand*
L_1	A, B, C	*Textured*
L_2	D, E, H, I, N	*Smooth*
L_3	F, G, J, K, Y, M, O	*Smooth*
L_4	P, Q, R	*Brand*

non-periodic and periodic scheduling are addressed, respectively. Additionally, a new case is analysed using formulation $M_2 - M_{2b}$ – to further extend the *HR* resource utilization rate for a given time horizon.

The models were implemented in GAMS (GAMS Rev 237 WIN-VS8 23.7.3 x86/MS Windows) and solved through CPLEX with a Intel Xeon X5680 at

3.33 GHz with 24 GB RAM. A time limit of 3600 CPU seconds was established in accordance with the maximum time allowed by the production manager to define a new schedule.

4.2 Optimisation Results

Non-periodic Operation Mode

Model M1 considers the non-periodic formulation of the scheduling case for the profit maximization, using the weekly production plan presented in Table 1 and a time horizon of 40 h. The optimal solution obtained is able to accomplish the demand plan of 196 kL, for a total profit of 15,600 m.u. with 4 *HR* elements, which represents a significant result improvement when compared to production records (161 kL and 6 *HR*). In the Gantt chart of Fig. 4, the schedule illustrates the sequence of tasks of each production orders and the resource allocation of the manufacturing machines [M_i] and filling lines [L_i]. For example, M_6 execute seven *Preparation* tasks for five different product orders, and the corresponding *Filling* tasks take place in lines L_1 and L_2 (after proper Storage task). With 4 elements, the utilization rate of the HR team for the proposed solution is 88 %. Figure 5 displays de availability of the *HR* resource and storage tanks (T_F, T_S and T_T groups) for the time horizon. In both cases, the higher availability verified in the begging of the horizon tend to fade with the increasing allocation to production tasks.

Periodic Operation Mode

Acknowledging that, on occasion, the company performs a campaign production for an extended horizon, a periodic scheduling was considered. Regarding the formulation detailed in model M2 and the production plan of 196 kL displayed in Table 1, the scheduling problem considers the horizon H of 3 identical cycles of 40 h T. The model generates an optimal solution with a profit of 15,600 m.u. with 4 *HR* elements, supplying the total demand with a schedule displayed in the Gantt chart of Fig. 6. The availabilities of the *HR* resource and storage tanks for a single cycle T are displayed in Fig. 7.

This formulation allows the overlap of some tasks taking place on the beginning/end of the planning horizon. In the results of M2, only storage tasks overlap the consecutive horizons (*Sto_I* and *Sto_G*). However, this aspect enables the scheduling problem to consent additional production capacity. For that reason, an additional model run – case M2b – was performed to analyse the maximization of the *HR* utilization rate for a cycle T. Considering the optimal results obtained in case M2 and setting accordingly the available manpower to 4 HR elements, the initial production plan of 196 kL is now extended. In order to restrain the computational complexity of the periodic formulation, only some additional orders were added to the production plan by duplicating orders B, G, I, J, M and P of Table 1. The Gantt chart displayed in Fig. 8 shows the schedule solution, maximizing the total

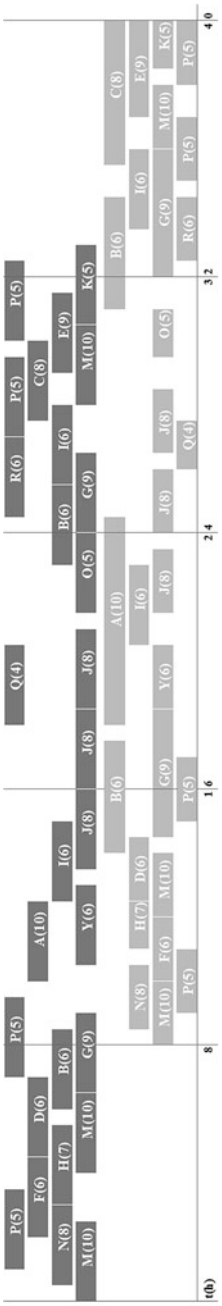


Fig. 4 Gantt chart of the model results M1

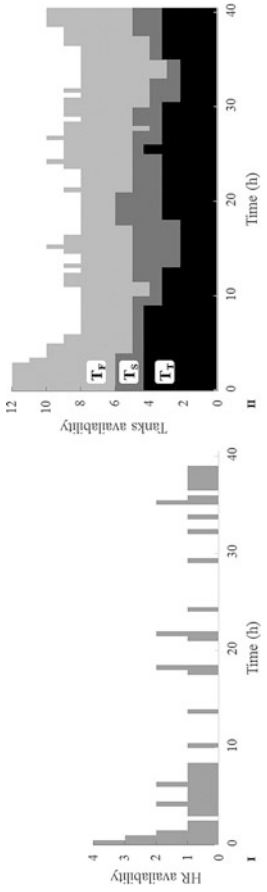


Fig. 5 HR (I) and storage tanks (II) availability for model M1

Fig. 6 Gantt chart of the model results M2

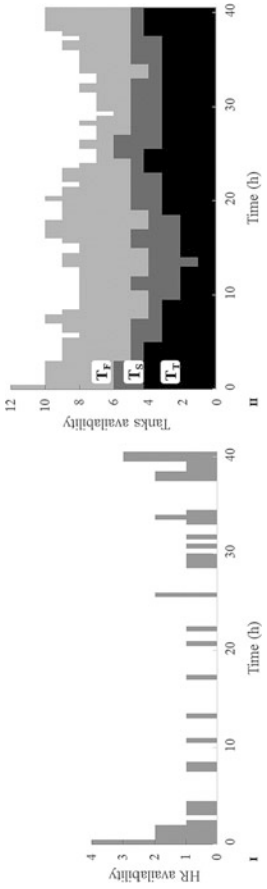


Fig. 7 HR (I) and storage tanks (II) availability for model M2

Fig. 8 Gantt chart of the model results M2b

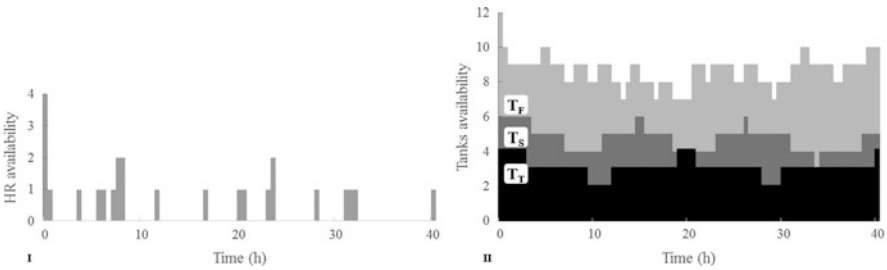


Fig. 9 HR (I) and storage tanks (II) availability for model M2b

Table 3 Main results of optimisation cases

	M1	M2	M2b	Real production records
Production Plan	196 kL	196 kL	207 kL	161 kL
Profit (m.u.)	15,600	15,600	16,700	10,100
HR elements	4	4	4	6
HR utilization rate	88 %	88 %	93 %	–

Table 4 GAMS models statistics

	Single equations	Single variables	Discrete variables	Relative gap (%)	Resource usage (s)
M1	28.121	15.362	4.320	0,0	15,1
M2	29.016	15.362	4.320	0,0	578,6
M2b	29.101	15.553	4.374	1,9	3.600,0

profit to 16,700 m.u., producing two additional orders I(6) and P(5) for a total plan of 207 kL. This approach enhances the utilization rate of the *HR* resource with 4 elements to 93 %. Figure 9 displays the availability of *HR* resource and storage tanks for the cycle *T*. Both solutions – cases M2 and M2b – provide significant improvements when compared with production records, despite the increasing mathematical complexity of the formulation M2b, which wasn't able to reach the optimal solution in the 3600 s time limit imposed.

Table 3 summarizes the main results and Table 4 the model characteristics of the analysed optimisation cases.

5 Conclusions

In this work, two MILP formulations, M1 and M2, based on the RTN framework were developed to solve a real scheduling problem of the paint industry. M1 and M2 addressed the non-periodic and periodic mode of operation of a production process, respectively. An additional study (M2b) was performed to analyse the

maximization of the utilization rate of resource *HR*. The sequence of production orders, processing tasks and allocated resources was detailed, maximizing the total profit. All model results revealed improved schedule solutions when compared with real production records for similar conditions, increasing the total profit in more than 54 % with the optimisation of the allocated manpower to only 4 elements. Only Model M2b wasn't able to determine an optimal solution within the time limit (1,9 % optimisation gap), since the high number of production orders and time horizon discretization increased the combinatorial nature of the problem. The periodic formulation suggested a more accurate formulation for the problem under study with the ability to perform an extended campaign for a wider horizon. The results of the case-study have provided important information to support the optimal management of the process schedule, with special relevance to the production output, resources availability and required manpower. Further analysis to process conditions not yet addressed, such as setup times or storage constraints, should also be considered to extend the problem formulation. Noteworthy, it is acknowledged that the RTN framework potentiates a uniform formulation for the scheduling of complex systems and a comprehensively problem solving.

Acknowledgements The authors would like to acknowledge the financial support of Fundação para a Ciência e Tecnologia under the grant SFRH/BD/51594/2011.

References

1. Barbosa-Póvoa, A.P.: A critical review on the design and retrofit of batch plants. Comput. Chem. Eng. **31**(7), 833–855 (2007)
2. Castro, P., Barbosa-Póvoa, A.P., Matos, H., Novais, A.Q.: Simple continuous-time formulation for short-term scheduling of batch and continuous processes. Ind. Eng. Chem. Res. **43**(1), 105–118 (2004)
3. Chen, C.L., Chang, C.Y.: A resource-task network approach for optimal short-term/periodic scheduling and heat integration in multipurpose batch plants. Appl. Therm. Eng. **29**(5), 1195–1208 (2009)
4. Chibeles-Martins, N., Pinto, T., Barbosa-Póvoa, A., Novais, A.Q.: A meta-heuristics approach for the design and scheduling of multipurpose batch plants. Comput. Aided Chem. Eng. **28**, 1315–1320 (2010)
5. Kondili, E., Pantelides, C.C., Sargent, W.H.: A general algorithm for short-term scheduling of batch operations-I MILP formulation. Comput. Chem. Eng. **2**, 211–227 (1993)
6. Méndez, A.C., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M.: State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. **30**(6–7), 913–946 (2006)
7. Moniz, S., Barbosa-Póvoa, A.P., Pinho de Sousa, J.: Regular and non-regular production scheduling of multipurpose batch plants. Comput. Aided Chem. Eng. **30**, 767–771 (2012)
8. Pantelides, C.C.: Unified frameworks for optimal process planning and scheduling. In: Foundations of Computer-Aided Process Operations, pp. 253–274. Cache publications, New York (1994)
9. Pinto, T., Barbosa-Póvoa, A., Novais, A.Q.: Optimal design and retrofit of batch plants with a periodic mode of operation. Comput. Chem. Eng. **29**(6), 1293–1303 (2005)

10. Pinto, T., Barbosa-Póvoa, A., Novais, A.Q.: Design of multipurpose batch plants: a comparative analysis between the STN, m-STN, and RTN representations and formulations. Ind. Eng. Chem. Res. **47**(16), 6025–6044 (2008)
11. Pinto, T., Barbosa-Póvoa, A., Novais, A.Q.: Design and scheduling of periodic multipurpose batch plants under uncertainty. Ind. Eng. Chem. Res. **48**, 9655–9670 (2009)
12. Shah, N., Pantelides, C.C., Sargent, R.W.H.: Optimal periodic scheduling of multipurpose batch plants. Ann. Oper. Res. **42**(1), 193–228 (1993)
13. Shaik, M.A., Vooradi, R.: Unification of STN and RTN based models for short-term scheduling of batch plants with shared resources. Chem. Eng. Sci. **98**, 104–124 (2013)
14. Simaria, A.S., Gao, Y., Turner, R., Farid, S.S.: Designing multi-product biopharmaceutical facilities using evolutionary algorithms. Comput. Aid. Chem. Eng. **29**, 286–290 (2011)
15. Verderame, P.M., Elia, J.A., Li, J., Floudas, C.A.: Planning and scheduling under uncertainty: a review across multiple sectors. Ind. Eng. Chem. Res. **49**(9), 3993–4017 (2010)
16. Vieira, M., Pinto-Varela, T., Barbosa-Póvoa, A.P.: Scheduling batch processing using the RTN discrete time formulation: a case study. In: 16th Congress of the Portuguese Association of Operations Research, Bragança, Portugal, pp. 378–385 (2013)