Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2011
26–30 June 2011.

Assessment of a Hybrid Approach for Nonconvex Constrained MINLP Problems

Florbela P. Fernandes[1], M. Fernanda P. Costa[2] and Edite M.G.P. Fernandes[3]

[1] *Department of Mathematics–Polytechnic Institute of Bragança, & CMAT–University of Minho, Portugal*

[2] *Department of Mathematics and Applications, University of Minho, Portugal*

[3] *Algoritmi R&D Center, University of Minho, Portugal*

emails: `fflor@ipb.pt`, `mfc@math.uminho.pt`, `emgpf@dps.uminho.pt`

Abstract

A methodology to solve nonconvex constrained mixed-integer nonlinear programming (MINLP) problems is presented. A MINLP problem is one where some of the variables must have only integer values. Since in most applications of the industrial processes, some problem variables are restricted to take discrete values only, there are real practical problems that are modeled as nonconvex constrained MINLP problems. An efficient deterministic method for solving nonconvex constrained MINLP may be obtained by using a clever extension of Branch-and-Bound (B&B) method. When solving the relaxed nonconvex nonlinear programming subproblems that arise in the nodes of a tree in a B&B algorithm, using local search methods, only convergence to local optimal solutions is guaranteed. Pruning criteria cannot be used to avoid an exhaustive search in the search space. To address this issue, we propose the use of a genetic algorithm to promote convergence to a global optimum of the relaxed nonconvex NLP subproblem. We present some numerical experiments with the proposed algorithm.

Key words: mixed-integer programming, branch-and-bound, genetic algorithm.
MSC 2000: 02.60.Pn

1 Introduction

A wide range of applications in the industrial processes are modeled as nonconvex constrained mixed-integer nonlinear programming (MINLP) problems, due to the restrictions imposed on some problem variables to take only integer values. In particular, one may find applications which include gas network problems, nuclear core reloaded problems, cyclic scheduling trim-loss optimization in the paper industry, synthesis problems, layout problems, thermal insulation systems. Other examples of this kind of problems appears in engineering designs, in metabolic pathway engineering or in molecular design (see for instance [1, 2, 4, 14, 19]).

In this paper we consider a mixed-integer nonlinear program in the following form:

$$\begin{array}{rl} \min & f(x,y) \\ \text{subject to} & g_j(x,y) \geq 0,\ j \in J \\ & x \in X,\ y \in Y \end{array} \tag{1}$$

where $X = \{x \in \mathbb{R}^n : l_x \leq x \leq u_x\}$ with $l_x, u_x \in \mathbb{R}^n$ and $Y = \{y \in \mathbb{Z}^p : l_y^1 \leq y \leq u_y^1\}$ with $l_y^1, u_y^1 \in \mathbb{Z}^p$. $f : \mathbb{R}^{n+p} \to \mathbb{R}$ and $g : \mathbb{R}^{n+p} \to \mathbb{R}^m$ are continuously differentiable functions, J is the index set of inequality constraints, and x and y are the continuous and discrete/integer variables, respectively. If the objective function f is convex and the constraint functions g_j are concave, the problem is known as convex, otherwise the problem is a non-convex MINLP [3].

It is known that nonconvex MINLP problems are the most difficult since they combine all the difficulties arising from the mixed-integer linear programming and non-convex constrained nonlinear programming (NLP). Taking into account that this kind of problems appears very frequently in industrial processes, it is fundamental to develop solution techniques to efficiently solve nonconvex constrained MINLP problems [8, 18, 22].

Therefore, the goal of this study is to analyze and propose a method for nonconvex constrained MINLP problems. The proposed hybrid method combines two strategies: a Branch-and-Bound (B&B) method to find integer solutions and a genetic algorithm (GA) type method to promote convergence to global solutions of the relaxed nonconvex NLP subproblems.

The paper is organized as follows. In Section 2, the proposed hybrid method is presented and B&B and GA methods are briefly described. Section 3 presents numerical results for 21 MINLP benchmark problems from the open literature and some conclusions are drawn. Finally, Section 4 presents the major conclusions and future work. The set of test problems used in this work is listed in the appendix.

2 The proposed hybrid method

To solve nonconvex constrained MINLP problems, a B&B-type method is used. Initially developed to solve combinatorial optimization problems, the B&B strategy has evolved to a method for solving more general problems, like for example the MINLP (1). B&B computes lower and upper bounds on the optimal value of f over successively refined partitions of the search space. The generated partition elements are saved in a list. Then they are selected for further processing and partition. The partition elements are deleted when their lower bounds are no lower than the best known upper bound for the problem. While branching on a binary variable creates two subproblems with that variable fixed in both problems, branching on a continuous variable in nonlinear programming may require an infinite number of subproblems. Furthermore, the relaxed

NLP subproblem that appears at each node of the B&B tree search may be nonconvex and it may be difficult to solve to global optimality. Classical gradient-based or even derivative-free local search methods may fail in solving nonconvex NLP problems. Thus, the herein proposed methodology for solving nonconvex MINLP uses a heuristic to solve the relaxed nonconvex NLP subproblems of the B&B tree search.

Existing methods for global optimization can be classified into two categories [15]: stochastic methods and deterministic methods. Stochastic methods sample the objective function for a number of points, with an outcome that is random. They are particularly suited for problems that possess no known structure that can be exploited, and in general do not require derivative information. In these methods, a probabilistic convergence guarantee can be provided. The genetic algorithm is an example of a population-based stochastic method. On the other hand, deterministic methods exploit analytical properties of the problem to generate a sequence of points converging to a global solution. They typically provide a mathematical guarantee for convergence to a minimum in a finite number of steps. The B&B method is a deterministic method.

In this paper, a new methodology to solve problem (1), based on a deterministic method and on a stochastic population-based method is presented. It relies on a B&B scheme and uses a genetic algorithm to promote convergence with a high probability to a global optimum of the relaxed nonconvex NLP subproblem (that arises in each node of the tree in the B&B algorithm). A brief description of the two strategies combined in the herein proposed hybrid method is presented below.

2.1 Branch-and-Bound method

B&B, originally devised for MILP (Mixed Integer Linear Program), can be applied to mixed-integer nonlinear problems too. The reader is referred to one of the first references to nonlinear Branch-and-Bound [9] and also to MINLP problems (see [16, 17] and the references therein included).

The B&B methodology can be explained in terms of a tree search. Initially, all integer variables are relaxed and the resulting relaxed NLP subproblem is solved. If all integer variables take an integer value at the solution then this solution also solves the MINLP. Usually, some integer variables take non-integer values. In that case, a tree search is performed in the space of the integer variables. The B&B algorithm selects one of those integer variables which take non-integer value and branches on it. Branching generates new NLP problems by adding simple bounds respectively to the new relaxed NLP subproblems. After that, one of these new subproblems is selected and solved.

The solution of each subproblem provides a lower bound for the subproblems in the descent nodes of the tree. This process continues until the lower bound exceeds the best known upper bound, the NLP subproblem is infeasible, or the solution provides integer values for the integer variables. The integer solutions (at the nodes of the tree) give upper bounds on the optimal integer solution. This process stops when there are no more nodes to explore.

2.2 The Genetic Algorithm

To solve the nonconvex MINLP problem the herein proposed approach combines the B&B method (described above) and the genetic algorithm, which is used to solve each relaxed nonconvex NLP subproblem that appears in the nodes of the B&B tree. The relaxed NLP subproblem assumes that all variables are continuous and has the form (according to the MINLP model (1)):

$$\begin{array}{rl} \min & f(z) \\ \text{subject to} & g_j(z) \geq 0, \ j \in J \\ & l \leq z \leq u \end{array} \tag{2}$$

where $z = (x, y)$, $z \in \mathbb{R}^m$ and $m = n + p$. Further, the lower and upper bounds of the problem satisfy: $l = \left(l_x, l_y^i\right)$, $u = \left(u_x, u_y^i\right)$ with $i \geq 1$ an integer index and $[l_y^i, u_y^i] \subset [l_y^{i-1}, u_y^{i-1}]$. The used notation means that the index i refers to a problem that is in a descent node of the problem $i - 1$.

A variety of techniques have been proposed to handle inequality constraints in NLP problems. The most widely used techniques rely on penalty functions. Here we are interested in a particular case, known as Lagrangian barrier function. Using the Lagrangian approach to solve the NLP problem in the form (2), a subproblem is formulated by combining the objective function and the nonlinear constraint functions in the barrier function Θ as follows:

$$\Theta(z, \lambda, s) = f(z) - \sum_{j \in J} \lambda_j \, s_j \log(g_j(z) + s_j). \tag{3}$$

where $\lambda_j \geq 0$ represents an estimate of the Lagrange multiplier associated with the constraint $g_j(z) \geq 0$ $(j \in J)$ and the components s_j of s are positive and are known as shifts. The general problem (2) is solved by a sequential minimization of the function (3) within the region defined by the simple bounds [7]. The method based on this Lagrangian barrier $\Theta(z, \lambda, s)$ proceeds by fixing λ to some estimate of the optimal Lagrange multipliers and s to some positive estimate of the initial slack variables, and then finding a value of z that approximately minimizes Θ. This new iterate z is then used to update λ, and then s, and the process is repeated. The vector of shifts s depends iteratively on the value of the multiplier vector and on a classical penalty parameter. In order to solve the problem (2), the following general algorithmic framework, as presented in Algorithm 1, is considered [7].

When the subproblem in Step 1 of Algorithm 1 is minimized to a required accuracy (verified in Step 2 of the algorithm), the Lagrangian multipliers are updated. This leads to a new function $\Theta(z, \lambda, s)$ and a new simple bound minimization problem. These steps are repeated until convergence to the optimal solution of (2) is achieved. A detailed description of the algorithm is shown in [6, 7, 13].

This paper is concerned with the use of the genetic algorithm to solve the subproblem that appears in Step 1 of Algorithm 1. The GA may be viewed as an evolutionary process wherein a population of solutions evolves over a sequence of iterations. Genetic algorithm selects individuals at random from the current population to be parents and

Algorithm 1 (Lagrangian barrier method)

Step 0.	Given initial estimates for z and initial estimates for λ and for penalty parameter. Set $k_o = 0$.
Step 1.	Compute shifts s. Use GA to compute z by solving $\min_z \Theta(z, \lambda, s)$ subject to $l \leq z \leq u$.
Step 2.	Test for convergence: Stop, go to Step 3 or go to Step 4.
Step 3.	Update Lagrange multiplier estimates. Maintain the penalty parameter. Increase k_o and go to Step 1.
Step 4.	Reduce the penalty parameter. Maintain the multiplier estimates. Increase k_o and go to Step 1.

uses this individuals to produce the children for the next generation. Over successive iterations and using the common operations of selection, crossover, mutation and scaling, the population "evolves" toward an optimal solution [13]. The basic structure of GA can be summarized as in Algorithm 2.

Algorithm 2 (GA algorithm)

Step 0.	Create a random initial population. Set $k_i = 1$.
Step 1.	Evaluate population using fitness Θ.
Step 2.	While stopping criteria are not satisfied: Select solutions for next population based on their fitness. Perform crossover and mutation. Accept new generation. Evaluate population using fitness Θ. Increase k_i and go to Step 2.

As far as the termination criteria are concerned, GA stops if one of the following conditions holds: the number of iterations exceeds a maximum threshold; the CPU time exceeds a maximum threshold (in seconds); or the cumulative change in the fitness Θ over stall iterations is less than or equal to a function tolerance.

3 Numerical Results

The proposed method (hereafter called BBGA) was implemented in `Matlab` and uses a B&B method combined with the GA method – *ga* function from the Optimization Tolbox of `Matlab`. The *ga* function is called inside the B&B algorithm (to solve the relaxed NLP in each node of the tree)and was run using the default options.

A collection of MINLP problems with inequality constraints and simple bounds is used to analyze the practical behaviour of BBGA. A comparison between this study, a previous work and other results in the literature is presented. The set of tested problems is displayed in the appendix. The problems are denoted by P1, P2, P2a, P3, $\cdots$, P20. For each problem the solver was run 30 times. All experiments were run on a

HP 2230s computer with an Intel(R) Core(TM)2 Duo CPU P7370 2.00GHz processor and 3 GB of memory.

To analyze the behaviour of the BBGA method and to perform a comparison with other results, Table 1 lists all the problems tested in this study "Prob.", the optimal solutions known in the literature,"Optimal f", and the related references, "Ref".

Table 1: Solutions obtained from the literature.

Prob.	**Ref**	**Optimal f**	**Prob.**	**Ref**	**Optimal f**	**Prob.**	**Ref**	**Optimal f**
P1	[10]	6.83325E-13	P7	[10]	-227.766	P14	[21]	-32217.4
P2	[10]	8.0839E-16	P8	[10]	0.0	P15	[12]	-17
P2a	[10]	6.66341E-11	P9	[21]	4.5796	P16	[11]	-8.5
P3	[10]	-0.35239	P10	[20]	2.00	P17	[20]	-5.68
P4	[10]	1.07746E-12	P11	[21]	2.1247	P18	[20]	2.00
P5	[10]	-0.407462	P12	[21]	1.076543	P19	[20]	3.4455
P6	[10]	-18.0587	P13	[21]	3.557473	P20	[20]	2.20

Table 2: Numerical results obtained for MINLPs with simple bounds.

				Average				
Prob.	m	p	**% success**	**time (s)**	**f eval.**	**Best f^***	**Average f^***	σ
P1	2	1	100	0.8	5779	4.930E-15	2.986E-08	2.885E-08
P2	2	1	13	1.0	7291	4.497E-10	1.86656E-08	1.754E-08
			87	1.0	7293	7.659E-11	1.7296E-08	1.751E-08
P2a	4	1	17	1.3	7965	3.984E-08	2.967E-05	3.158E-05
			50	1.1	7765	1.574E-08	1.010E-05	3.159E-05
P3	2	1	67	0.6	4642	-0.35239	-3.524E-01	5.106E-05
P4	2	1	13	1.0	7802	6.692E-13	1.323E-09	9.551E-10
			87	1.1	8403	2.700E-11	4.383E-07	2.136E-06
P5	2	1	33	1.2	4369	-0.40746	-4.075E-01	1.997E-06
			53	1.3	4492	-0.40746	-4.075E-01	4.879E-08
P6	2	1	13	1.4	5271	-18.0587	-1.806E+01	1.582E-07
			83	1.6	6289	-18.0587	-1.806E+01	9.678E-04
P7	2	1	17	2.1	8161	-227.766	-2.278E+02	8.874E-04
			70	2.0	7680	-227.766	-2.278E+02	2.249E-03
P8	2	1	100	0.8	5829	3.659E-10	5.127E-08	1.536E-07

The obtained results with BBGA method are presented in two separate tables. Table 2 contains the results with the problems with simple bounds; Table 4 contains the results from the problems with inequality constraints. In Table 2 and Table 4, the best objective function value "Best f^*" obtained over the 30 runs is reported for each test problem. In order to show more details concerning the quality of the obtained solution, the average "Average f^*" and the standard deviation "σ" of the best obtained function values are also reported in both tables. Moreover, success rates of obtaining the global minimum "% success", the average numbers of CPU time "time" (in seconds) and function evaluations "f eval.", are shown in columns 4–6. It is also reported, in columns 1–3, the name of the problem, the total number of variables, "m", and the number of integer variables, "p".

From Table 2 it is possible to observe that the set of tested problems covers different situations: problems with small size, problems with or without the same number of integer/continuous variables and problems with one or more than one global minimizer.

As it can be seen, BBGA method has a success rate superior to 85% for 7 problems, out of 9 problems. It is noteworthy that the BBGA method has a success rate equal to 100% for 4 problems. The accuracy of the achieved solutions, in terms of "Best f^*" is very high when compared to the solutions reported in Table 1; the standard deviations of the f values are close to zero.

Table 3 summarizes the results obtained, for this set of problems using a strategy based on a B&B scheme and a simulated annealing heuristic pattern search – BBSAHPS method (see [10] and the references included).

Table 3: Numerical results obtained for Problems P1 – P8 from [10].

	P1	P2	P2a	P3	P4	P5	P6	P7	P8
% success	46	36	87	87	19, 18	49	49	52	42
		26	3		6, 6	40	43	41	
Av. time (s)	0.1	0.1	0.3	0.1	0.1	0.0	0.1	0.1	0.1
			0.2						
Av. f eval.	991	744	1626	966	918, 842	555	610	643	656
		733	1615		859, 865	562	593	573	

It is possible to observe, from Table 2 and Table 3, that the computational cost in terms of CPU time required by the proposed BBGA method, as well as the required number of function evaluations are greater than those of BBSAHPS method. For problem P4, BBSAHPS is able to find four global minimizers, while BBGA finds only two global minimizers. On the other hand the successful rate of BBGA is much better than the successful rate of BBSAHPS: BBGA method has a success rate greater than 67% (problem P2a and P3) and equal to 100% for 4 problems while BBSAHPS has a successful rate less than 50% for three problems.

To address the solution of constrained mixed-integer nonlinear programs, a list of well-known problems, in particular, some taken from *minlplib* [5, 20] is used. The obtained results, with BBGA method, for constrained MINLP are reported in Table 4. The dimension of these problems range between 2 and 7 avriables. The number of inequality constraints range between 1 and 9. This set of problems provides a variety of small MINLP problems.

As it can be seen, BBGA method has a success rate superior to 87% for 8 problems, out of 12 problems. It is noteworthy that the BBGA method has a success rate equal to 100% for 5 problems. Some runs did not converge to the global minimum. For instance,

Table 4: Numerical results obtained for constrained MINLPs.

Prob.	m	p	% success	Average time (s)	Average f eval.	Best f^*	Average f^*	σ
P9	7	4	53	13.7	46678	4.5797	4.5987	2.86E-02
P10	2	1	100	2.5	10481	2.00	2.00	6.84E-05
P11	2	1	100	2.4	11527	2.1245	2.1246	2.08E-04
P12	3	1	20	3.3	13635	1.076864	1.078992	2.40E-03
P13	7	4	23	13.5	47282	3.557742	3.564265	7.63E-03
P14	5	2	100	1.3	5220	-32217.4	-32217.4	1.48E-11
P15	2	1	100	3.4	14292	-17	-17	1.80E-04
P16	2	1	97	6.0	25752	-8.5	-8.5	1.55E-04
P17	3	2	87	65.0	247055	-5.6848	-5.684E+00	1.941E-03
P18	4	2	67	3.2	12808	2.000	2.000E+00	8.920E-07
P19	2	1	100	7.8	23489	3.4455	3.446E+00	1.981E-05
P20	4	3	87	4.5	15290	2.20	2.200E+00	5.840E-05

problem P16 has 3% of the runs that have converged to a strong local minimum [11]. Exceptions are problems P12 and P13, which have a successful rate less than 50%. With all these problems the BBGA method could successfully find the global minimum, taking into account the integrality of some variables. The problem P17 requires a larger CPU time when compared with the other problems. This is related with the definition of the problem and the space of the integer variables.

The accuracy of the achieved solutions, in terms of "Best f^*", and for almost all problems, is very good and the consistency of our proposed method is high, since the standard deviations of the f values are close to zero. For problem P11, if a comparison is made in terms of "Best f^*", from Table 1, it is possible to observe that the solution of BBGA method is slightly better than those reported in the literature.

For problem P12 the objective value reported in Table 1 is slightly better than the one obtained with the BBGA. On the other hand, it is possible to observe that BBGA method needs much less functions evaluations than the algorithms used in [21]. In a general way, BBGA method needs much more function evaluations. This is due to the use of a GA method to solve the relaxed NLP problems in each node of the B&B tree search. P17 is the problem with the highest average time and average function evaluations.

Although the chosen problems are small-dimensional, it is important to emphasize that almost all the problems represent real applications of MINLPs.

4 Conclusions and future work

This paper presents a method, herein denoted by BBGA, which relies on a deterministic method and on a stochastic population-based method to solve nonconvex constrained MINLP problems. A Branch-and-Bound procedure is combined with a Lagrangian barrier-function-based genetic algorithm to find the minimum of relaxed nonconvex NLP problems.

The BBGA method was implemented using `Matlab` and some results are shown for 21 test problems, from which 12 are constrained MINLP problems. This method is able to find the global optimum of all problems with integer restrictions on some variables. A comparison between the results from BBGA and the results from the literature is presented. We may conclude that the performance of the BBGA method is quite satisfactory, since the results obtained with BBGA method are competitive with the results reported in the literature.

The reported CPU time is in general high. Thus, an improvement in the BBGA method in order to reduce computational requirements will be carried out in the future. Future developments will be focused, also, on solving sets of medium and large scale problems.

Appendix: Details of test problems

The collection of 21 test problems used in this study is listed below [10, 11, 12, 20, 21].

Problem 1 (P1)

$$\min \ (1.5 - x_1(1-x_2))^2 + (2.25 - x_1(1-x_2^2))^2 + (2.625 - x_1(1-x_2^3))^2$$
$$x_1 \in \{-5, \cdots, 5\}, \quad x_2 \in [-4.5, 4.5]$$

Problem 2 (P2)

$$\min \ (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$$
$$x_1 \in \{-10, \cdots, 10\}, \ x_i \in [-10, 10], \ i = 2 \ldots n,$$

(P2) for $n = 2$;

(P2a) for $n = 4$

Problem 3 (P3)

$$\min \ 0.25x_1^4 - 0.5x_1^2 + 0.1x_1 + 0.5x_2^2$$
$$x_1 \in [-10, 10], \ x_2 \in \{-10, \cdots, 10\}$$

Problem 4 (P4)

$$\min \ \cos^2(x_1) + \sin^2(x_2)$$
$$x_1 \in [-5, 5], \quad x_2 \in \{-5, \cdots, 5\}$$

Problem 5, 6, 7

$$\min \ 10^m x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-m}(x_1^2 + x_2^2)^4$$

(P5): $m = 1$ and $x_2 \in [-2, 2], x_1 \in \{-2, \cdots, 2\}$;

(P6): $m = 2$ and $x_2 \in [-4, 4], x_1 \in \{-4, \cdots, 4\}$;

(P7): $m = 3$ and $x_2 \in [-8, 8], x_1 \in \{-8, \cdots, 8\}$

Problem 8 (P8)

$$\min \ (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$
$$x_2 \in [-2, 4], \quad x_1 \in \{-2, \cdots, 4\}$$

Problem 9 (P9)

$$\min \ (y_1 - 1)^2 + (y_2 - 2)^2 + (y_3 - 1)^2 - \ln(y_4 + 1) + (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2$$

$$\begin{aligned}
\text{s. t.} \ & y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \le 5 \\
& y_3^2 + x_1^2 + x_2^2 + x_3^2 \le 5.5 \\
& y_1 + x_1 \le 1.2 \\
& y_2 + x_2 \le 1.8 \\
& y_3 + x_3 \le 2.5 \\
& y_4 + x_1 \le 1.2 \\
& y_2^2 + x_2^2 \le 1.64 \\
& y_3^2 + x_3^2 \le 4.25 \\
& y_2^2 + x_3^2 \le 4.64 \\
& x_1 \in [0, 1.2], x_2 \in [0, 1.281], \\
& x_3 \in [0, 2.062], \\
& y_1, y_2, y_3, y_4 \in \{0, 1\}
\end{aligned}$$

Problem 10 (P10)

$$\begin{aligned}
\min \ & 2x + y \\
\text{s. t.} \ & 1.25 - x^2 - y \le 0 \\
& x + y \le 1.6 \\
& x \in [0, 1.6], y \in \{0, 1\}
\end{aligned}$$

Problem 11 (P11)

$$\begin{aligned}
\min \ & -y + 2x - \ln(x/2) \\
\text{s. t.} \ & -x - \ln(x/2) + y \le 0 \\
& x \in [0.5, 1.4], y \in \{0, 1\}
\end{aligned}$$

Problem 12 (P12)

$$\begin{aligned}
\min \ & -0.7y + 5(x_1 - 0.5)^2 + 0.8 \\
\text{s. t.} \ & -e^{x_1 - 0.2} - x_2 \le 0 \\
& x_2 + 1.1y \le -1 \\
& x_1 - 1.2y \le 0.2 \\
& x_1 \in [0.2, 1], x_2 \in [-2.22554, -1], \\
& y \in \{0, 1\}
\end{aligned}$$

Problem 13 (P13)

$$
\begin{array}{ll}
\min & (y_1-1)^2+(y_2-1)^2+(y_3-1)^2-\ln(y_4+1)+ \\
 & (x_1-1)^2+(x_2-2)^2+(x_3-3)^2 \\
\text{s. t.} & y_1+y_2+y_3+x_1+x_2+x_3 \le 5 \\
 & y_3^2+x_1^2+x_2^2+x_3^2 \le 5.5 \\
 & y_1+x_1 \le 1.2 \\
 & y_2+x_2 \le 1.8 \\
 & y_3+x_3 \le 2.5 \\
 & y_4+x_1 \le 1.2 \\
 & y_2^2+x_2^2 \le 1.64 \\
 & y_3^2+x_3^2 \le 4.25 \\
 & y_2^2+x_3^2 \le 4.64 \\
 & x_1 \in [0,1.2]\,,\, x_2 \in [0,1.8]\,,\, x_3 \in [0,2.5]\,, \\
 & y_1, y_2, y_3, y_4 \in \{0,1\}
\end{array}
$$

Problem 14 (P14)

$$
\begin{array}{ll}
\min & 5.357854x_1^2+0.835689y_1x_3+37.29329y_1- \\
 & 40792.141 \\
\text{s. t.} & 85.334407+0.0056858y_2x_3+0.0006262y_1x_2- \\
 & 0.0022053x_1x_3 \le 92 \\
 & 80.51249+0.0071317y_2x_3+0.0029955y_1y_2+ \\
 & 0.0021813x_1^2-90 \le 20 \\
 & 9.300961+0.0047026x_1x_3+0.0012547y_1x_1+ \\
 & 0.0019085x_1x_2-20 \le 5 \\
 & x_1, x_2, x_3 \in [27,45]\,,\, y_1 \in \{78,\cdots,102\}\,, \\
 & y_2 \in \{33,\cdots,45\}
\end{array}
$$

Problem 15 (P15)

$$
\begin{array}{ll}
\min & 3y-5x \\
\text{s. t.} & 2y^2-2y^{0.5}-2x^{0.5}y^2+11y+8x \le 39 \\
 & -y+x \le 3 \\
 & 2y+3x \le 24 \\
 & x \in [1,10]\,,\, y \in \{1,\cdots,6\}
\end{array}
$$

Problem 16 (P16)

$$
\begin{array}{ll}
\min & -x-y \\
\text{s. t.} & xy-4 \le 0 \\
 & x \in [0,4]\,,\, y \in \{0,\cdots,8\}
\end{array}
$$

Problem 17 (P17)

$$
\begin{array}{ll}
\min & -0.00201x_1^4x_2x_3^2 \\
\text{s. t.} & -675+x_1^2x_2 \le 0 \\
 & -0.419+0.1x_1^2x_3 \le 0 \\
 & x_1, x_2 \in \{1,\cdots,200\}\,,\, x_3 \in [0.1,0.2]
\end{array}
$$

Problem 18 (P18)

$$
\begin{array}{ll}
\min & 2+4x_3^2+2x_4+2x_1+2x_2 \\
\text{s. t.} & -x_3+3x_4-5 \le 0 \\
 & 2x_3-x_4-5 \le 0 \\
 & -2x_3+x_4 \le 0 \\
 & x_3-3x_4 \le 0 \\
 & -6x_1+x_3 \le 0 \\
 & -5x_2+x_4 \le 0 \\
 & x_1, x_2 \in \{0,1\}\,,\, x_3 \in [0,6]\,,\, x_4 \in [0,5]
\end{array}
$$

Problem 19 (P19)

$$
\begin{array}{ll}
\min & 1.1\left((2x_1-10)^2+(x-2-5)^2\right)+ \\
 & \sin\left((2x_1-10)^2+(x-2-5)^2\right) \\
\text{s. t.} & 0.7x_1+x_2 \le 7 \\
 & 2.5x_1+x_2 \le 19 \\
 & x_1 \in [0,10]\,,\, x_2 \in \{0,\cdots,10\}
\end{array}
$$

Problem 20 (P20)

$$
\begin{array}{ll}
\min & 5x_1^2+x_2+x_3+x_4 \\
\text{s. t.} & 3x_1-x_2-x_3 \le 0 \\
 & -x_1+0.1x_3+0.25x_4 \le 0 \\
 & 2-x_2-x_3-x_4 \le 0 \\
 & 2-x_2-x_3-2x_4 \le 0 \\
 & x_1 \in [0.2,1]\,,\, x_2, x_3, x_4 \in \{0,1\}
\end{array}
$$

References

[1] K. Abhishek, S. Leyffer and J. Linderoth, *FilMINT: An Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs*, Technical Report ANL/MCS-P1374-0906, Mathematics and Computer Science Division, Argonne National Laboratory (2006).

[2] K. Abhishek, S. Leyffer and J. T. Linderoth, *Modeling without categorical variables: a mixed-integer nonlinear program for the optimization of thermal insulation systems*, Optimization and Engineering **11** (2010) 185–212.

[3] L. Biegler and I. Grossmann, *Retrospective on optimization*, Computers and Chemical Engineering **28** (2004) 1169–1192.

[4] P. Bonami, L. Biegler, A. Conn, G. Cornuejols, I. Grossmann, C. Laird, J. Lee,A. Lodi, F. Margot, N. Sawaya and A. Wachter, *An algorihmic framework for convex mixed integer nonlinear programs*, Discrete Optimization **5** (2008) 186–204.

[5] Michael R. Bussieck, Arne Stolbjerg Drud and Alexander Meeraus, *MINLPLib–A Collection of Test Models for Mixed-Integer Nonlinear Programming*, INFORMS Journal on Computing **15(1)** (2003) 114–119.

[6] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds*, SIAM Journal on Numerical Analysis **28(2)** (1991) 545–572.

[7] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *A Globally Convergent Augmented Lagrangian Barrier Algorithm for Optimization with General Inequality Constraints and Simple Bounds*, Mathematics of Computation **66(217)** (1997) 261–288.

[8] R. H. Byrd, J. Nocedal and R. A. Waltz, *KNITRO: An Integrated Package for Nonlinear Optimization in Large-Scale Nonlinear Optimization*, G. di Pillo and M. Roma (Eds.) (2006) 35–59.

[9] R.J. Dakin, *A tree search algorithm for mixed integer programming problems*, Computer Journal **8** (1965) 250–255.

[10] F. P. Fernandes, M. Fernanda P. Costa and Edite M. G. P. Fernandes, *A Deterministic-Stochastic Method for Non-convex MINLP Problems*, Proceedings of 2nd International Conference on Engineering Optimization, H. Rodrigues et al. (Eds.), (2010) CD-ROM (1198) .

[11] C.A. Floudas, *Nonlinear and Mixed-Integer Optimization Fundamentals and Applications*, Oxford University Press, New York, 1995.

[12] C.A. Floudas, P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gumus, S. T. Harding, J. L. Klepeis, C. A. Meyer and C. A. Schweiger, *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, London, 1999.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.

[14] I. Grossmann, *Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques*, Optimization and Engineering **3** (2002) 227–252.

[15] J. Lee, *A novel three-phase trajectory informed search methodology for global optimization*, Journal of Global Optimization **38** (2007) 61–77.

[16] S. Leyffer, *Deterministic Methods for Mixed Integer Nonlinear Programming*, PhD Thesis, University of Dundee, 1993.

[17] S. Leyffer, *Integrating SQP and branch and bound for mixed integer nonlinear programming*, Computational Optimization and Applications **18** (2001) 295–309.

[18] S. Leyffer, A. Sartenaer and E. Wanufelle, *Branch-and-Refine for Mixed Integer Nonconvex Global Optimization*, Preprint ANL/MCS-P1547-0908, Mathematics and Computer Science Division, Argonne National Laboratory, 2008.

[19] S. Leyffer, J. Linderoth, J. Luedtke, A. Miller and T. Munson, *Applications and Algorithms for Mixed Integer Nonlinear Programming*, J. Phys.: Conf. Ser. **180** (2009) 1–5.

[20] *MINLPLib–A Collection of Test Models for Mixed-Integer Nonlinear Programming*, `http://www.gamsworld.org/minlp/minlplib/minlpstat.htm` (2011).

[21] T. W. Liao, *Two hybrid differencial evolution algorithms for engineering design optimization*, Applied Soft Computing **10** (2010) 1188–1199.

[22] I. Nowak and S. Vigerske, *LaGO: a (heuristic) branch and cut algorithm for nonconvex MINLPs*, Central European Journal of Operations Research **16** (2008) 127–138.