SMART AND SUSTAINABLE MANUFACTURING SYSTEMS FOR INDUSTRY 4.0

Edited by
Vijaya Kumar Manupati
Goran D. Putnik
Maria Leonilde Rocha Varela

CRC Press

Conceptual Multi-agent System Design for Distributed Scheduling Systems

Filipe Alves[a,b]*, *Ana Maria A. C. Rocha*[a], *Ana I. Pereira*[b] *and Paulo Leitão*[b]

[a]*ALGORITMI Center, University of Minho, 4710-057, Braga, Portugal.*

[b]*Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal.*

E-mail: filipealves@ipb.pt*, arocha@dps.uminho.pt, apereira@ipb.pt, pleitao@ipb.pt.

Abstract

With the progressive increase in the complexity of dynamic environments, systems require an evolutionary configuration and optimization to meet the increased demand. In this sense, any change in the conditions of systems or products may require distributed scheduling and resource allocation of more elementary services. Centralized approaches might fall into bottleneck issues, becoming complex to adapt, especially in case of unexpected events. Thus, Multi-agent systems (MAS) can extract their automatic and autonomous behaviour to enhance the task effort distribution and support the scheduling decision-making. On the other hand, MAS is able to obtain quick solutions, through cooperation and smart control by agents, empowered by their coordination and interoperability. By leveraging an architecture that benefits of a collaboration with distributed artificial intelligence, it is proposed an approach based on a conceptual MAS design that allows distributed and intelligent management to promote technological innovation in basic concepts of society for more sustainable in everyday applications for domains with emerging needs, such as, manufacturing and healthcare scheduling systems.

Keywords: Multi-agent System, Distributed Scheduling, Decision-making.

1. Introduction

Nowadays, the decision-making process based on planning and scheduling are very important in the manufacturing domain and industries, economics, health, and service management. In the ongoing competitive environment, the processes scheduling has become crucial to survive in the quotidian services (Michael, 2018). The goal of scheduling is to allocate different resources to tasks over time in order to optimize one, or more objectives, for example, increase the execution speed, reduce the task runtime, minimize communication delay, communication cost, and emergences or priority problems, among others (Baker and Trietsch, 2013; Leung, 2004; and Michael, 2018).

The scheduling problems are complex and difficult to solve, considering their large number of integrated entities and their interactions, therefore they are classified in the literature as NP-hard (Pinedo *et al.,* 2015). The scheduling is often encountered in practice, being over the years studied in the operations research literature (Cardoen *et al.*, 2010; Leung, 2004; Michael, 2018; and Van den Bergh *et al.*, 2013). Furthermore, the dynamic nature of the scheduling environments makes difficult to obtain optimal solutions. Hence, several approximation methods (i.e., heuristics, meta-heuristics, or agent-based approaches) are some of the proposals to deal with scheduling problems in real environment (Chan and Chung, 2013).

A well-prepared schedule can significantly improve the production, lead-times, utilization, due-date-related measures and improve the profitability of end products or services. The globalization on production systems allowed to make them more reconfigurable, flexible and demand driven. Centralized scheduling approaches have become obsolete since the activities and resources are controlled using just a single decision maker, due they are less responsive to unexpected events, inflexible to emergencies, and unable to respond to the dynamic needs of the market (Leitão, 2009). By drawing a line between the current works and our system, our goal follows the determination of development and implement a complex system that demonstrates, in practice (deploy in real-world problems), the benefits of distributed scheduling (DS) subject to unexpected events in dynamic environments. However, organizations and companies have moved from a centralized to a decentralized architecture, due to the changing in the business world (Behnamian and Ghomi, 2016). In practice, what distinguishes the DS system from centralized models is the use of system decomposition, since its focus on the communication scheme will allow solving sub-problems in different domains that can be combined to form final scheduling. For example, the subcontractors of a major company, who may have alternative processing abilities, would typically prepare their schedules independently from one another, although consistent with the full schedule. According to highly distributed environment, local decision-makers are competitors with individual objectives. The need for making scheduling decisions in decentralized systems has given rise to a new area, which is denoted as distributed scheduling. Distributed scheduling is an approach in which smaller parts of a scheduling problem are solved by local decision makers (local entities) who may have conflicting objectives or specific orders and who coordinate their sub solutions through certain communication mechanisms to achieve the general goals of the system. Hence, the lower level of the organizational hierarchy delegates and solve locally and independently the decision problems, by different entities and/or resources of the system. These entities can be multiprocessors in a communication network, different departments of a company, flexible manufacturing systems or for example, health care routing or allocation belonging to a national health system.

In a distributed way, decision-making may increase system responsiveness that is very important for scheduling applications, where a short-term decision requires up-to-date information and data to generate feasible and practical schedules. This work aims to contribute to the progress of the literature review in the area of scheduling, especially in the development of intelligent mechanisms that enable innovative and optimized solutions, reorganization, and fast responses to change conditions in the dynamic scheduling. More specifically, the following objectives are defined for the design and specification of a conceptual and intelligent approach that allows to increase the scheduling operating in a decentralized mode, allowing to maintain high levels of optimal solutions through optimization methods and, at the same time, to reduce the operation costs with fast responses of MAS. With these advanced mechanisms, namely distributed scheduling, prioritization, dynamic and fast response, it will be possible to achieve desired adaptation and optimization of task elements in order to achieve self-scheduling or rescheduling in reactive or unexpected events with minimal degradation of the quality of the solutions. The architectural design will be based on an intelligent control approach based on the swarm system and MAS design principles, combined with optimization methods that provide dynamic adaptation and optimization capabilities to address the issues where each agent is enhanced with, self-organizing oriented mechanisms, communication abilities and control capabilities to retrieve, analyze and forecast data in order to obtain the required information to improve the framework and deploy in real environments of scheduling. DS can be seen as allowing to create schedules by local decision makers considering all the objectives, restrictions, and constraints within the overall system goals. In terms of the research question, the critical decisions in a DS system, it is possible to observe and identify critical design factors, such as the whole system, individual agents, and inter-agent relations. According to the general perspective of the system: it will be possible to define how the scheduling problem will or can be decomposed, generate an independent decision-making, have independent decision-makers assigned to sub-problems and for example, enable group decision-making. According to the view of individual agents, it will be possible to identify what information prevails in each of the agents, possible trade-offs between objectives, and how the solution can be updated to eliminate some conflicts.

The paper is organized as follows: Section 2 overviews the contribution on the main topic and their literature review. Section 3 describes the preliminary MAS System Approach as architecture design and Section 4 presents the preliminary results for the validation methodology and some discussion and critical view. Finally, Section 5 presents and rounds up the conclusions and future work ideas.

2. Literature review

Traditionally, classical optimization algorithms are used to solve scheduling problems, considering them, normally, as static and deterministic. However, for many optimization problems of industrial interest, the computation of optimal solutions by analytical methods it is almost impossible (Bartz-Beielstein et al., 2004). Thus, the adaptation of ideas from various areas, mainly based on nature, has resulted in the development of optimization methods, such as natural-inspired or evolutionary metaheuristics. As a result, using metaheuristics to attain a near optimal solution in a reasonably shorter period is more realistic than using traditional analytical approaches. However, the dynamic and stochastic nature of industrial environments increases the complexity of the problem, demanding more adaptive and efficient handling of planned or unplanned events in real time, such as, changing orders, priority orders, resources breakdown, worker’s illnesses, and large repairs. It is also important to consider the complexity associated with the ramp-up phase of complex and highly customized services, which are exceptionally challenging for allocation, planning, scheduling, and control. Taking this into consideration, new methods and Information and Communication Technologies (ICT) are required to develop strategies to respond more quickly to unexpected events, creating novel and emerging approaches to the scheduling problem. In this sequence, complex systems need to adapt their processes with the help of flexible and dynamics infrastructures. For that reason, scheduling perspectives take into account the latest research in the domain of Artificial Intelligence (AI) and software engineering, especially in the area of MAS, optimization methods and swarm intelligence which are increasingly designed and specified approaches for distributed ICT (Lezama *et al.,* 2019). Swarm intelligence is a derived concept that can be defined as "*the emergent collective intelligence of groups of simple and single entities*" (Bonabeau et al., 1999), reflecting the emergent phenomenon that occurs without a predefined plan and not driven by a central entity. With this approach, the swarm principles can be used in a distributed scheduling architecture providing a network of schedulers, organized as a swarm, and each one is responsible for the schedule in an organizational unit of an enterprise or specific service, namely a factory, cell, health unit, among others.

The increased complexity of architectures and problems in computer science required the development of parallel programming and distributed programming to take advantage of new software and hardware architectures. Distributed Artificial Intelligence (DAI) was born within AI to meet the requirements for the development of complex distributed systems. The MAS paradigm is one example of DAI (Ferber, 1999), composed of an intelligent society, based on entities, especially cooperatives and autonomous, called agents. These entities coordinate and interact according to their tasks and activities, making use of their knowledge and skills, in order to achieve their goals in a global way. This paradigm can provide innovative management, design and modelling for complex, dynamic and heterogeneous distributed systems (Weiss, 1999). In contrast with traditional centralized and hierarchical approaches that split the problem into hierarchically dependent functions, MAS performs its activities in parallel, thus characterized by its decentralization.

The agent concept does not have a consensual or unique definition mainly due to its features and attributes. Some proposed definitions that can be found in the literature are:

- "an agent is a persistent computation that can perceive its environment and reason and act both alone and with other agents. The key concepts in this definition are interoperability and autonomy" (Singh, 1998).
- "an agent is a computational entity that can be viewed as perceiving and acting upon its environment, that is autonomous and that operates flexibly and rationally in a variety of environmental circumstance" (Weiss, 1999).

- “an agent is a computer system that is situated in an environment and that is capable of autonomous action in this environment in order to meet its design objectives” (M. Wooldridge, 2009).

Independently of its definition, an agent is characterized by its ability to act and adapt flexibly, responding effectively to changes in the environment by sensing the environmental changes in states. It also has autonomy and pro-activity that allows him to take initiative to meet its own goals (M. J. Wooldridge and Jennings, 1995). Thus, normally an abstract representation of an agent presents it equipped with sensors and actuators, which are responsible for interacting with the environment as well as with a reasoning mechanism, responsible for determining what actions the agent should perform according to their perceptions and their internal states.

The agents are characterized by a set of properties that determine its behaviour. The following are the main properties that an agent must have (M. Wooldridge, 2009; and M. J. Wooldridge and Jennings, 1995):

- Autonomy – operations without human intervention, or systems, getting control of their actions and internal states.
- Social ability – interactions with other agents to achieve their goals, through specific languages and communication protocols, which allow them to negotiate and cooperate rather than simply exchange information.
- Reactivity – ability to observe the surrounding environment and being empowered to respond as changes in the system arise.
- Pro-activity – ability to display goal-driven behaviours, taking the initiative to reach their goals, rather than simply act in response to the environment stimulus.

A single agent in some cases can be enough to control, monitor or carry out the task’s execution. However, some tasks need to be performed by more-than-one agent due to their size and/or complexity, giving rise to the concept of MAS. Each agent, despite the need to meet its own goals, can act on behalf of different users or perform tasks to meet the purposes of an application. To succeed in their interactions, some capabilities are required (Russell and Norvig, 2002):

- Cooperation – the ability to work among them to achieve common objectives.
- Coordination – the ability to manage the inter-dependencies between activities, for example, to use non-shareable resources, or to perform a sequence of tasks.

- Negotiation – the ability to have communication mechanisms, such as bidding (e.g., Contract-Net protocol (Smith, 1980)). The negotiation mechanism has the ability to make agreements on issues of common interest, for example, by an offer and a counteroffer of the involved parties.

This interaction is supported by the usage of a communication language, namely agent communication language (ACL) and/or by a set of interaction protocols. There are two major ACL, namely the KQML - Knowledge Query and Manipulation Language (Finin *et al.,* 1994) and also the Foundation for Intelligent Physical Agents based on Agent Communication Language (FIPA-ACL) (Labrou *et al.,* 1999). Both languages use the theory of speech acts to provide semantics to messages through a set of speech acts, called performatives (M. Wooldridge, 1997). The performatives are used by agents to represent or interpret their requests and intentions, thus, when a message is received, an agent can understand the sender intentions and decide what to do. For example, some performatives specified by FIPA-ACL are: *Inform* – used to indicate that the agent is communicating information or a fact; *Request* – used to indicate that the agent is requesting a service or information; *Agree* – indicates that the agent agrees to a request from another agent; *Not Understood* – used to indicate that the agent did not understand the received message. While the agent communication language specifies the messages of agents, the interaction protocols specify the sequence of messages exchanged between two or more agents in a given scenario, such as, for an information or service request, as well as for negotiation and cooperation.

In recent years, several agent platforms and frameworks have been developed (Bordini *et al.,* 2006; Kravari and Bassiliades, 2015). The purpose of these tools is to simplify the development of agents to support developers with agent's basic infrastructure, such as agent's communication, agent's negotiation, agent's collaboration, and other features inherent of the MAS approach. Generally, the agent platforms offer a set of built-in features for the development and execution of agent-based approaches. In order to ensure interoperability between heterogeneous agents developed on different platforms, many standards and specifications for the agent technology were developed. In this context, FIPA[1] is characterized by being an organization that specifies and defines a set of standards developed with the intention to promote

[1] http://fipa.org/

interoperability between heterogeneous agents. FIPA standards are widely accepted by the community and there are several platforms and compatible frameworks, such as JADE[1]. There are also frameworks and platforms developed exclusively for agent-based simulation systems, such as NetLogo[2].

Therefore, MAS has multidisciplinary applications, such as, dynamic product (Wang *et al.*, 2016), manufacturing control (Monostori *et al.*, 2006), production planning (Trentesaux *et al.*, 2013), logistics (Jabeur *et al.*, 2017), health (Nealon, 2003) or home health care (Marcon *et al.*, 2017), and many others (Leitão *et al.*, 2012).

3. MAS System Approach

This section aims to explore and specify a conceptual MAS-based architecture that intelligently searches for, selects, and composes distributed scheduling solutions.

In this sense, and considering the requirements of scheduling problems, their characteristics and restrictions, the proposal of a conceptual MAS should take into account all assumptions and, if possible, contain some of the principles of swarm intelligence, which may facilitate the distributed scheduling structure. Swarm principles or guidelines, generally refers to solving a problem with an ability that emerges from the interaction of simple information-processing units. The concept of swarm suggests multiplicity, distribution, stochasticity, randomness, and disorder. In this context, the swarm concepts are a new computational and behavioural paradigm for solving distributed problems based on self-organization applied, for example, to scheduling. On the other hand, most of the previous scheduling strategies and approaches consider scheduling as static, deterministic, and normally centralized. However, several domains of scheduling are subject to a dynamic environment, with new jobs continuously arriving to the system, certain resources becoming unavailable, disruptions and additional resources or orders introduced. The swarm approach offers plenty of powerful mechanisms to handle emerging and evolving environments, where complex systems are built upon entities that exhibit simple behaviours and have reduced cognitive abilities (betting on the hypothesis of a

[1] http://jade.tilab.com/

[2] https://ccl.northwestern.edu/netlogo/

swarm approach). In fact, everybody knows that "*a single ant or bee isn't smart but their colonies are*" (Miller, 2007), and also that they are capable of exhibiting very surprising complex behaviours. In such environments, the coordination of activities uses simple feedback coordination mechanisms in opposite to the traditionally rigid centralized control (betting on the hypothesis of a decentralized approach).

To design a main scheme of the swarm, for more adaptive an efficient scheduling solution, optimization intelligence capabilities, it is necessary to design these tools for several domains. For this reason, each DS system can be considered as a network of schedulers, as shown in Figure 1 (generally case). Each distributed scheduler is responsible for the schedule of an organizational unit of a service or company, namely a factory, workshop or station, or a health unit belonging to a network of Health Units.

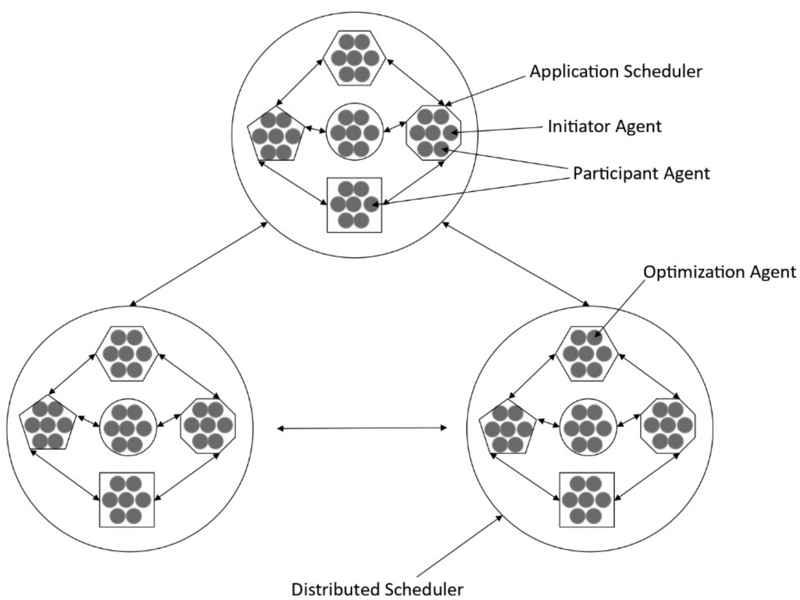

Figure 1. Recursive MAS architecture for a distributed scheduling.

In this manner, a swarm is a network of individual schedulers, $S = \{S^{(1)}, S^{(2)}, S^{(3)}, \ldots, S^{(n)}\}$, where the global schedule emerges from all the interactions, among these schedulers. In this sense, complex systems, like scheduling, are built upon entities that exhibit simple behaviours and have reduced cognitive abilities. For example, in the concept of the swarm, it is found in the literature simple entities, which are based on principles of extreme utility, such as, on the fact that they are regulated by simple rules, they have mainly reactive behaviour and usually interact with each other and with their environment, without the existence of a central authority.

The individual schedulers may be constantly interacting with each other to achieve the global (emergent) scheduling, exchanging information on the events that occur during the process tasks and could affect the order execution of the others. The swarm of schedulers is connected to the legacy systems, namely databases and ontology services, enabling the schedulers to retrieve required data from these sources, based on an example on an exchange data model (explained later). To design something similar to this, it is important to know how to define swarms to balance the interests of selfish individuals and groups of individuals. In this sense, some strategies can be implemented, namely:

- The scheduling problem may be divided into several smaller scheduling problems according to a logical dependency, e.g., a workshop scheduler can comprise several station schedulers.
- The scheduling problem may be divided in a way that different schedulers can be used to search faster alternative solutions in parallel, e.g., using different scheduling algorithms or exploratory searches (optimization methods).

Both situations reinforce the importance of using the swarm principles in the design of these scheduling systems. The specification of individual schedulers is independent of the swarm principles and can be implemented using different algorithms. In addition, individual schedulers can be implemented using multi-agent technology, composed by a set of software agents that represent interests of orders, resources, products, workers, operations, and materials. In the last approach, the agents interact with each other to achieve the overall schedule through proper negotiation mechanisms, and it is possible to combine with optimization methods through developed extensions to communicate and integrate different platforms. In this strategy, an individual scheduler may be composed by a network of entities, which are able to produce the desired scheduling, where entities are related by a set of relations. The swarms of heterogeneous schedulers are supported in a transparent manner, contributing for a smooth migration from old solutions using MAS technology combined with optimization methods. For this purpose, the proper definition of communication interfaces is required, as the mean for data exchange between individual schedulers.

The previously presented swarm principles allow DS to obtain dynamic solutions using different types of distributed and optimized methods. This process facilitates the system design, where each entity can be any application scheduler, i.e., initiator, participants, and optimization agent. This allows to design a complex system using different types of agents, in particular:

- Initiator Agent - represents the “jobs” (a job represents a set of operations, processes patients, ...) that provide sequence and dynamic coordination to generate the process of the goals according to the scheduling and participants agents.
- Participant Agent - represents the participants of the solution (machines, health professionals, etc.). Can be responsible to generate the tasks and forecast the future generation of scheduling. They will manage their activities and behavior, considering their priorities, needs, current status and forecast of future demand. In addition, they are subject to unexpected events, interruptions, or failures. For this purpose, this agent also considers the implementation of scheduling, system reorganization capabilities, prioritization, and optimization.
- Optimization Agent - represents the functional agents, that is, pieces of software used to carry out sub-tasks such as search or working strategies, local optimization, or scheduling management. For example, multi-criteria, decision methods, reasoning or even AI can perform certain functions and act as a mediator.

Each agent is responsible to collect and analyze their data from their connected software (or a set of text files associated to the problem). This data exchange between the different agents supplies the decision-making abilities of each one, that can improve the optimization procedures and scheduling activities.

In this way, the agent's conceptual architecture is illustrated in the Figure 2, a proposal for applications in real-world environment.

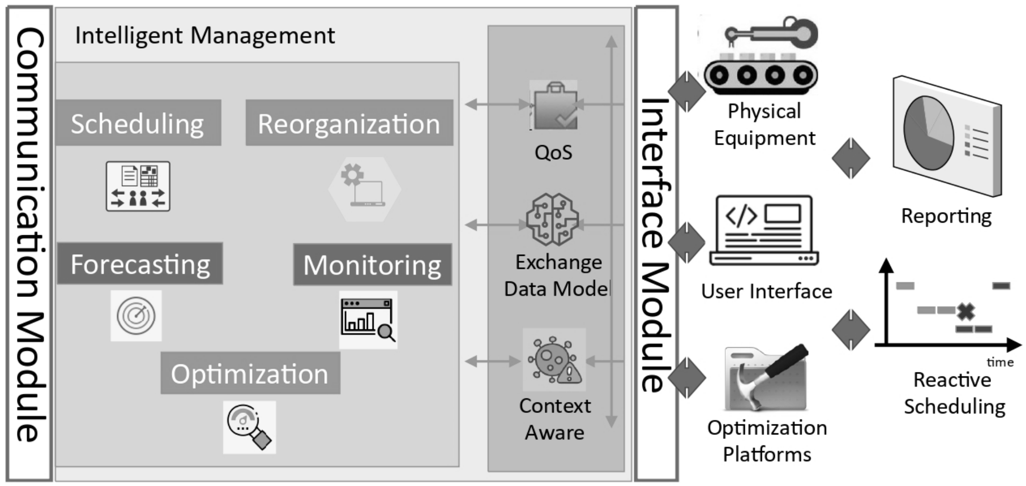

Figure 2. Architecture for intelligent management of an agent (adapted from (Alves *et al.*, 2019c)).

The agents will be dynamically started through the system administration using communication to exchange and access information, namely with the use of a database as an engine. The multi-agent system will use cloud computing to decentralize effort tasks and later distribute intelligence based on a communication module, that will manage the exchange and access of information. The smart functions relate analysis of the data from the set of tasks belonging to the DS, comprising:

- Scheduling - assesses the possibilities and performs the tasks allocations (set of operations or processes) and management (participants involved) for each entity.
- Forecasting – forecasts the generated scheduling by using the current and historical management and scheduling data, as well as the disruptions or failures information.
- Reorganization - represents the reactive scheduling facing immense pressures and dynamic factors, such as changes in processing time, emergencies, prioritization and/or participants failures. The idea is to apply the act of modifying the offline scheduling in response to disruptions or identification of the most important and critical tasks for each entity and time horizon, present a predictive-reactive state that a reschedule can start in response to a specific event, depending on the current context. It will be possible to consider, for example, new orders, failures, and unexpected events.
- Monitoring - monitors the current situation and the decision-making evolution to detect the efficiency, reliability, flexibility, especially in dynamic procedures, such as the start

of scheduling and prioritization or in some cases, initiate the coordination and dynamics for self-scheduling in an attempt disruption or failures.

- Optimization - functions to promote the decentralized system and scheduling generation that require cooperation and optimization in the use of shared resources. Optimization will be responsible to find optimal or near-optimal solutions for local or global schedules applying specific and evolutionary optimization methods.

The design of such adaptive and self-reconfigurable scheduling systems also considers reorganization concepts to support the regulation of the dynamics of such a complex network of a swarm of schedulers. Reorganization can be seen in two perspectives: structural (changing the relationships among the individual entities) and behavioural (changing the internal behaviour of individual entities). In another perspective, structural self-organization appears when:

- Dynamic reconfiguration of the scheduling problem occurs, e.g., by adding/removing tasks, resources, or workers, or by changing the dependencies between the individual schedulers inside the swarm scope.
- A dynamic reorganization of individual entities, involved in exploratory searches for planning or scheduling solutions, occurs, aiming to achieve better emergent solutions.

The intelligent management will address three small assistance and support modules, which in turn will communicate with the interface module, in a bi-directional approach. Relatively to the exchange data model, a common process model is required, such as the entering of input data and the obtained solution (output) to the problems caused by an architecture and connections, which includes different problems, constraints, restrictions, and methods. On the other hand, the interface module can be connected to physical devices, optimization platforms (e.g., dedicated platforms/algorithms embedded) and access a dashboard dedicated to monitoring and interact with specific optimization platform or some communication service.

The optimization platforms module is responsible for the agent behaviour and system performance, provided by dedicated platforms embedded with algorithms and optimization methods, such as MatLab, ILOG Cplex or Python. Thus, it will be possible to obtain cooperation and communication between some platforms, especially dedicated to specific actions, enhancing their resources and libraries.

Resulting from the combination of MAS and optimization methods, it is needed to standardize the data format, its semantics, and/or variables. Figure 3 illustrates a general model of data exchange for any application of scheduling, in an attempt to standardize the data and the consequent exchange of information.

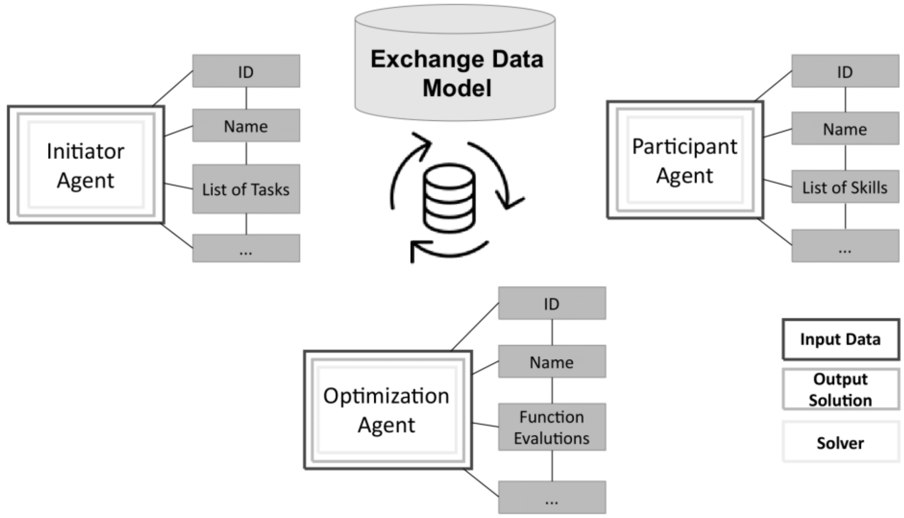

Figure 3. General exchange data model.

Each agent will have at his disposal the data model capable of being applied in different domains. In turn, the conceptual structure will be supported depending on the type of agent and will be developed with heterogeneous perspectives in the construction data sources for dynamic environments (Figure 4).

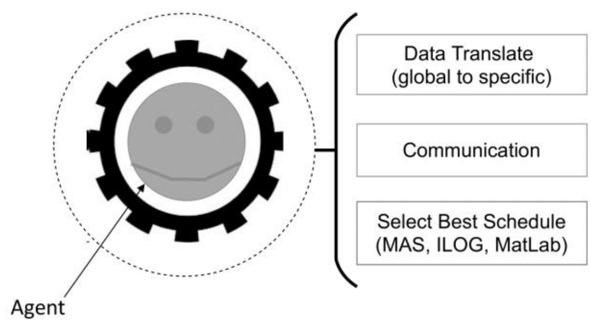

Figure 4. Capabilities of each agent.

In this sense, the modules inside each agent will allow not only to adjust their own parameters but also to adjust according to the result of the interaction with the other agents. This means that each agent has the ability to communicate, interact and cooperate with other agents, using in this case the communication module, following proper coordination patterns that specify how and which interaction pattern should be used based upon a specific context.

The communication protocol will be, mainly, from agent to agent, using preferentially the

FIPA-ACL protocol. In this way, each entity can select specific agents and interact with them. Figure 5 represents generic interactions between the aforementioned agents. These interactions will be modelled using Agent Unified Modelling Language (AUML) (Bauer *et al.*, 2001) that extends the UML to design agents through an intuitive graphical representation of the architecture and processes.

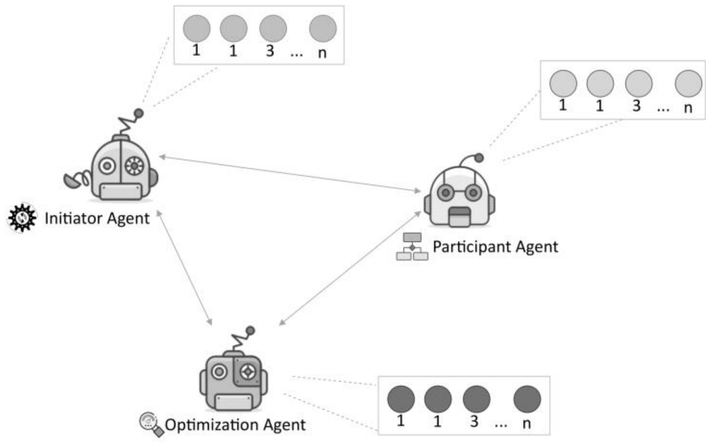

Figure 5. Interactions among agents.

The intra-agent interface module is responsible to provide mechanisms that make transparent the access to the optimization platforms, e.g., MatLab or other software, supporting the collection of data and the execution of commands in the distributed system, e.g., the scheduling of tasks set.

In general, the proposed disruptive architecture will empower the decision-making for DS in real applications with dynamics environments, providing competitive and intelligent technological skills in reporting cases and strategic and automated planning for reactive scheduling problems.

4. Case Studies

The proposed approach can be applied to different domains aiming to provide environments to validate the proposal benefits. Two application domains are considered for this study: one in the manufacturing area and the other in the Home Health Care (HHC) area. At this stage, it is not clear which domain can benefit the most from this work. Therefore, the following methodology seeks to validate the conceptual MAS approach, as illustrated in Figure 6.

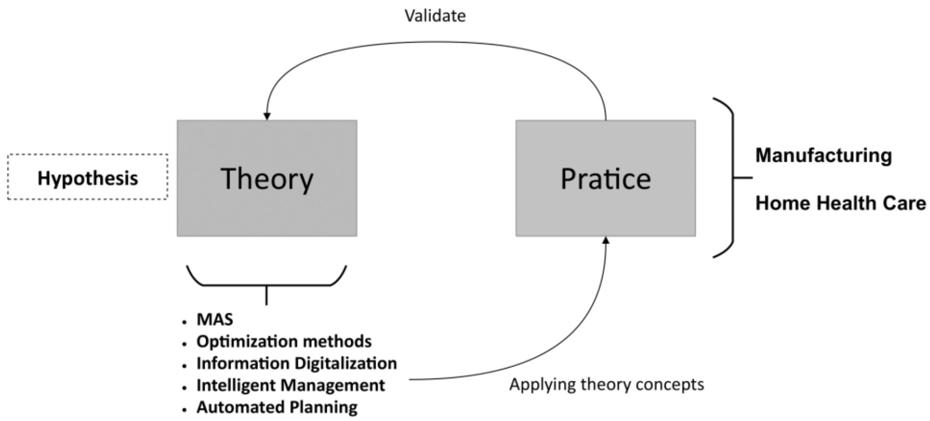

Figure 6. Validation methodology.

4.1 Manufacturing

The case study based on the manufacturing domain, namely the flexible manufacturing system production cell "AIP PRIMECA" (Trentesaux *et al.*, 2013), has a set of several robotic workstations. A set of operations (i.e., jobs) are performed on the workstations, generating different products that in turn, have a production workflow according to their sequence of operations until they obtain the final product for the consumer. In turn, these products can be performed on different workstations depending on their skills, however, each station has different performances. The goal concerns the reduction of the re-planning time considering the increase in the quality of the services provided, under very dynamic constraints, e.g., the occurrence of unexpected events or the need for reconfiguration. Obviously, these problems fit into several fields of research, such as critical path analysis, supply chain management, resource allocation, among others. Note that the idea is to avoid using centralized approaches, but instead, to use innovative optimization methods provided by combining with MAS. For this case study, it is useful to understand what (and how) our proposal can offer for this particular case, enabling a resolution of problems with:

- Enable and prioritize task effort distribution.
- Reduce the possibilities of interrupted processes.
- Faster and possibly more accurate reaction to condition changes.

This case study intends to highlight the advantages of promoting a reactive and flexible architecture for distributed scheduling, seeking, whenever possible, to balance the workload of those involved through a conceptual and simulated model that evaluates, in runtime, several replacement options and proposals for new scheduling in case of necessity. As new proposals emerge, the best ones will be selected and will support the agent's organization.

Some approaches and applications have already been developed in the manufacturing domain, involving the "AIP PRIMECA" and similar cases. The dataset was generated using NetLogo, a multi-agent programmable modelling environment, powered by their agent-based simulations and properly validation for the case study (Alves *et al.*, 2018b; and Alves *et al.*, 2019d). The experimental results achieved dynamic responsiveness for distributed scheduling with fast responses to changes in the environment and support decision-making in an interactive and user-friendly way.

4.2 Home Health Care

A home health care (HHC) support service allows the provision of care (medical, nursing, or social) at the home of its users (Fikar and Hirsch, 2017). The HHC has gained importance in recent years, particularly in Portugal, as it allows users to travel less and more comfort in their treatment (of particular importance given the aging of the population). The management of an HHC has a strong component related to the adequate use of resources (doctors, nurses, social workers, vehicles) in the fulfilment of the users' treatments (typically characterized in spatial and temporal terms and also by the specific requirements of the caregivers). The focus is to enable decision-making at an operational level, where as a rule it is up to the health professional who will provide the HHC to ensure the schedule and transport conditions for it, which currently does not have optimized circuits of visits or any computational support. In the current HHC management system (usually manual) it is often necessary to make decisions based on priority principles of care, leaving aside some care due to lack of time or excessive workload for healthcare professionals. In addition, the frequency of some HHC does not respect the regularity imposed, due to flaws in the scheduling channels and the lack of time that professionals have for proper planning. Resource management, such as vehicles and routes, is often scarce and hospital or health unit's management becomes regulated in terms of access and proper functioning, even in case of unexpected events that require real-time action (Cissé *et al.*, 2017). Typically, traffic jams, road accidents, weather conditions, new orders and emergencies should be considered when designing a conceptual architecture capable of supporting a complex optimization system, with several constraints and restrictions (Rasmussen *et al.*, 2012).

The main idea is to consider the problem of assigning tasks outside in a Health Unit, belong to the National Health System, or applied in HHC belong to social solidarity institutions. The approach will be in a first instance, it will be to modernize the current HHC system, digitizing the information and enabling an information system with intelligent and optimized decision making. In a second phase, it will be a priority to find the optimal scheduling for visits and home care circuits, especially, considering complex professional's assignments, routes, and workload balancing, for a given day, week, or specific time horizon. In addition, the maintenance and monitoring of correct continuity of care will be predominant. Therefore, it will be important to apply the proposed methodology in an attempt to decompose the problem into sub-problems (e.g., division in Health Units, daily and/or monthly scheduling, traveling issue, workload balancing) and provide distributed schedules with local optimization, to be inserted and included in the overall scheduling or allocation.

Some approaches and applications in the HHC domain, have already been developed, involving a health unit, in a real implementation environment with a specific real dataset (Alves *et al.*, 2019a; Alves *et al.*, 2019b; and Alves *et al.*, 2017). However, in recent times, studies point to the need to introduce intelligent systems with fast reaction in dynamic environments, mainly due to the lack of flexibility of the most classical optimization techniques. For that reason, MAS has already been introduced to take advantage of cooperation and intelligent mechanisms to decentralize the system, which normally operates in static mode. In a first approach, a simulated methodology in NetLogo, provided an excellent capacity to generate good autonomous and coordinated solutions, contributing to a better decision making and usefulness (Alves *et al.*, 2018a). In the future, the idea is to contribute to an HHC system based on a digital and distributed ecosystem (platform/app), using different ICT-based solutions, sources, tools, and techniques.

4.3 Evaluation and Performance Measurements

Regardless of the case study, the proposed approach should be evaluated and validated, in relation to some main aspects and features. The comparison of different existing approaches and applications comprises a very complex task since they use to present some specific functionalities, purposes and other aspects that can require too much effort and time to obtain the comparing measurements. Therefore, regarding the comparison between the proposed approach and other existing approaches and applications, it is intended to perform a more qualitative evaluation. In addition, quantitative assessments intend to be carried out through the analysis and measurement of some behaviours of the case study system prototype. The evaluation process will consist in the assessment and quantification of the results derived from the system prototype execution in the case study scenarios, considering as main metrics the response time, output accuracy and computational resource utilization or effort.

In this sense, since the proposed approach intends to cover the multi-agent system design for DS, the first aspect that should be evaluated is the local distributed vs. centralized monitoring, using as metrics the response time and monitoring output accuracy. In order to simplify the experiments, the computational resource requirements (such as processing capacity and memory), can be fixed during the quantitative analysis and evaluated in a qualitative manner, considering the input information, variables, and constraints.

The configuration of the local and distributed experiment will consist of several agents using communication mechanisms and optimization methods for collaboration and individually monitor different system components. The results of such configuration, in terms of response time and accuracy, should be compared with the results of a centralized configuration, where the data of all components should be sent to a central repository and then analysed to obtain a global schedule.

In this experiment, the distributed diagnosis should be achieved by the agents' interaction, while the centralized diagnosis will be performed by only one agent. Thus, the distributed approach is expected to require more network resources, while the centralized approach should require more memory. The accuracy of the diagnosis, presented by both approaches, should be equivalent and the time should not differ too much. In this sense, the distributed approach may be qualitatively better based on its flexibility and adaptability in autonomous scheduling.

In distributed scheduling, the society of agents will be able to receive requests and respond according to their specifications, seeking, whenever possible, to find and select the agents with the best proposals to be allocated the scheduling. The data should be compared with the initial schedule as one of the hypotheses that measures expectations about a specific feature. The main goal, in general, is to try to automatically compose the best planning of tasks, according to the requirements and objectives in question, making it possible to make statements by forwarding and receiving messages. Obviously, in case of experiments disruptions or emergencies, and the need of self-scheduling, the collected data should be analysed to be possible the decision, with minimal human intervention, on the feedback using qualitative indicators (e.g., flexibility, robustness, agility, etc.) and measure the following quantitative key performance indicators (KPI). Thus, some KPI can be measured, such as:

- Scheduling composition (maximum time, distance, or other objectives).
- Quality of composition (average quality of the scheduling composition played inside the swarm agents).
- Proposal's negotiation and agent utility (how good an agent can offer a schedule, and if the best offer is chosen).
- Average communication load (exchanged information in the system).
- Average number of agents (average number of agents required to compose a certain schedule or task request).
- Average forwarding messages (average cost of dispatching requests to other agents).
- Resilience of the network (ability to remove agents from the network without incurring in loss of performance).
- Average message size (exchange message between the agents).

In the Table 1, the set of evaluation intended to be performed is summarized:

Table 1. Summary of the proposed approach evaluation and metrics.

Description	**Metric**	**Expected Results**
Compare, in a qualitative manner, the proposed approach with other existing approaches and applications	Relevant functionalities and features required and desired to attend the current industrial and health issues and challenges	Cover most of the relevant aspects and features
Evaluate the monitoring and diagnosis tasks considering the proposed distributed swarm vs. a centralized approach	Response time, response accuracy, resource constraints	Obtain better response time, with acceptable accuracy, considering resource variables, constraints requirements
Compare, in a quantitative manner, the MAS	Scalability and flexibility (measure the performance of all previous KPI)	Cooperating and evolve in order to reduce and increase several points (e.g., decrease the schedule composition costs and improves the utility).
Evaluate solutions, runtime, and statistical parameters in the optimization methods	Parameters, execution time, statistical analysis, quality, computational effort of execution and robustness	Adjust the parameters, solving the problems in a reasonable time, guarantee the quality of the "optimal" solution and apply different problems and instances

The analysis of the solutions, composition, runtime, and statistical parameters will allow to obtain information about all the system, its modules, and procedures, to support decision-making. Thus, by using this information and measuring its impact on the overall system performance and how this has contributed to improving the dynamic adaptation, reconfiguration of processes, efficiency, effectiveness, trade-offs, and robustness of conceptual MAS for DS, will be a great approach to take into account.

5. Conclusions and Future Work

This work points out and discusses a MAS infrastructure for distributed scheduling and resource allocation, able to adapt and evolve using DAI (i.e., mechanism of relationships among agents) and fast decisions in dynamic and reactive environments. All the conceptual and technical aspects allow for a robust, intelligent, and powerful control approach characterized by the MAS capabilities to provide adaptation and fast responses to address the scheduling and allocation challenges for multidisciplinary domains.

Thus, the system is able to adapt and evolve using structural self-scheduling (i.e., mechanism of relationships among agents), optimized solutions and fast decisions in dynamic and reactive environments. Two case studies focusing distributed and dynamic environments were considered. The focus for the future will be on the experimental approach, in the attempt to potentiate an architecture for distributed scheduling resolution. Due to the dynamics and resources of the system concerned, performance measures and metrics will be crucial for the evaluation of the prototype.

It is expected a robust, intelligent, and powerful control approach characterized by the combination of MAS and optimization methods with swarm capabilities to provide adaptation and cooperation to address the scheduling challenges. In addition, intelligent and resilient mechanisms, namely forecasting and self-scheduling, will offer the possibility of developing and testing algorithms and strategies for a context-aware supervision and monitoring that can contribute to a more effective system adaptation. Distributed scheduling can be specified by a decentralized approach with intelligent services and procedures to support the lack of computational availability and reactive mechanisms.

The conceptual MAS design for DS system showed that it is a promising approach, but still, the validation of the current implementation is mandatory as future work to extrapolate the results and analyze the principles used in terms of evaluation and performance measures, namely the KPI. In this way, the DS system can improve existing results with responsiveness to dynamic data, decentralization, and optimization in domains with emerging needs, with great perspectives of applicability, flexibility, and scalability.

Summing up, it was proposed an approach to explore the digitization of information and cloud support for user-friendly use, will allow an exponential applicability and an aid to decision making in dynamic environments. The system idealized in a MAS approach with several intelligent capabilities and optimization algorithms will allow its portability and authenticity, in a scalable and flexible infrastructure for monitored management, even in real-time scheduling environments.

Acknowledgments. This work has been supported by FCT - Fundação para a Ciência e a Tecnologia within the R&D Units Projects Scope: UIDB/00319/2020 and UIDB/05757/2020. Filipe Alves is supported by FCT Doctorate Grant Reference SFRH/BD/143745/2019.

References

Alves, F., Alvelos, F. P., Rocha, A. M. A., Pereira, A. I., and Leitão, P., (2019a). Periodic vehicle routing problem in a health unit. *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems – ICORES*, pp. 384-389.

Alves, F., Costa, L., Rocha, A. M. A., Pereira, A. I., and Leitão, P., (2019b). A multi-objective approach to the optimization of home care visits scheduling. *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems – ICORES*, pp. 435-442.

Alves, F., Pereira, A. I., Barbosa, J., and Leitão, P., (2018a). Scheduling of home health care services based on multi-agent systems. *In International conference on practical applications of agents and multi-agent systems*, pp. 12-23.

Alves, F., Pereira, A. I., Fernandes, F. P., Fernandes, A., Leitão, P., and Martins, A., (2017). Optimal schedule of home care visits for a health care center. *In International conference on computational science and its applications*, pp. 135-147.

Alves, F., Rocha, A. M. A., Pereira, A. I., and Leitão, P., (2019c). Distributed scheduling based on multi-agent systems and optimization methods. *In International conference on practical applications of agents and multi-agent systems*, pp. 313-317.

Alves, F., Varela, M. L. R., Rocha, A. M. A., Pereira, A. I., Barbosa, J., and Leitão, P., (2018b). Hybrid system for simultaneous job shop scheduling and layout optimization based on multi-agents and genetic algorithm. *In International conference on hybrid intelligent systems*, pp. 387-397.

Alves, F., Varela, M. L. R., Rocha, A. M. A., Pereira, A. I., and Leitão, P., (2019d). A human centred hybrid mas and meta-heuristics-based system for simultaneously supporting scheduling and plant layout adjustment. *FME Transactions*, pp. 699-710.

Baker, K. R., and Trietsch, D., (2013). Principles of sequencing and scheduling. John Wiley & Sons.

Bauer, B., Müller, J. P., and Odell, J. (2001). Agent uml: A formalism for specifying multiagent software systems. *International journal of software engineering and knowledge engineering*, 11(03), pp. 207-230.

Behnamian, J., and Ghomi, S. F., (2016). A survey of multi-factory scheduling. *Journal of Intelligent Manufacturing*, 27(1), pp. 231-249.

Bonabeau, E., Marco, D. D. R. D. F., Dorigo, M., Théraulaz, G., & Theraulaz, G., (1999). *Swarm intelligence: from natural to artificial systems* (No. 1). Oxford university press.

Bordini, R. H., Braubach, L., Dastani, M., Seghrouchni, A. E. F., Gomez-Sanz, J. J., Leite, J., and Ricci, A., (2006). A survey of programming languages and platforms for multi-agent systems. *Informatica*, 30(1).

Cardoen, B., Demeulemeester, E., and Beliën, J., (2010). Operating room planning and scheduling: A literature review. *European journal of operational research*, 201(3), pp. 921-932.

Chan, H. K., and Chung, S. H., (2013). Optimisation approaches for distributed scheduling problems. *Taylor & Francis*.

Cissé, M., Yalçındağ, S., Kergosien, Y., Şahin, E., Lenté, C., and Matta, A., (2017). Or problems related to home health care: A review of relevant routing and scheduling problems. *Operations Research for Health Care*, 13, pp. 1-22.

Ferber, J., (1999). Multi-agent systems: An introduction to distributed artificial intelligence (1st ed.). USA: Addison-Wesley Longman Publishing Co., Inc.

Fikar, C., and Hirsch, P., (2017). Home health care routing and scheduling: A review. *Computers & Operations Research*, 77, pp. 86-95.

Finin, T., Fritzson, R., McKay, D., and McEntire, R., (1994). Kqml as an agent communication language. *In Proceedings of the third international conference on information and knowledge management*, pp. 456-463.

Jabeur, N., Al-Belushi, T., Mbarki, M., and Gharrad, H., (2017). Toward leveraging smart logistics collaboration with a multi-agent system based solution. *Procedia Computer Science*, 109, pp. 672-679.

Kravari, K., and Bassiliades, N., (2015). A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1), pp. 11.

Labrou, Y., Finin, T., and Peng, Y., (1999). Agent communication languages: The current landscape. *IEEE Intelligent Systems and Their Applications*, 14(2), pp. 45-52.

Leitão, P., (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering applications of artificial intelligence*, 22(7), pp. 979-991.

Leitão, P., Mařík, V., and Vrba, P., (2012). Past, present, and future of industrial agent applications. *IEEE Transactions on Industrial Informatics*, 9(4), pp. 2360-2372.

Leung, J. Y., (2004). Handbook of scheduling: algorithms, models, and performance analysis. CRC press.

Lezama, F., Palominos, J., Rodríguez-González, A. Y., Farinelli, A., and de Cote, E. M., (2019). Agent-based microgrid scheduling: An ict perspective. *Mobile Networks and Applications*, 24(5), pp. 1682-1698.

Marcon, E., Chaabane, S., Sallez, Y., Bonte, T., and Trentesaux, D., (2017). A multi-agent system based on reactive decision rules for solving the caregiver routing problem in home health care. *Simulation Modelling Practice and Theory*, 74, pp. 134-151.

Michael, L. P., (2018). Scheduling: theory, algorithms, and systems. Springer.

Miller, P., (2007). The genius of swarms. National Geographic, 212 (1), 126-147.

Monostori, L., Váncza, J., and Kumara, S. R., (2006). Agent-based systems for manufacturing. *CIRP annals*, 55(2), pp. 697–720.

Nealon, J. L., (2003). Applications of software agent technology in the health care domain. Springer Science & Business Media.

Pinedo, M., Zacharias, C., and Zhu, N., (2015). Scheduling in the service industries: An overview. *Journal of systems science and systems engineering*, 24(1), pp. 1-48.

Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J., (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3), pp. 598-610.

Russell, S., and Norvig, P., (2002). Artificial intelligence: a modern approach. Upper Saddle River, N.J.: Prentice Hall.

Singh, M. P., (1998). Agent communication languages: Rethinking the principles. *Computer*, 31(12), pp. 40-47.

Smith, R. G., (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Computer Architecture Letters*, 29(12), pp. 1104-1113.

Bartz-Beielstein, T., Parsopoulos, K. E., and Vrahatis, M. N., (2004). Analysis of particle swarm optimization using computational statistics. *In Proceedings of the international conference of numerical analysis and applied mathematics* (ICNAAM 2004), pp. 34-37.

Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P., and Barbosa, J., (2013). Benchmarking flexible job-shop scheduling and control systems. *Control Engineering Practice*, 21(9), pp. 1204-1225.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L., (2013). Personnel scheduling: A literature review. *European journal of operational research*, 226(3), pp. 367-385.

Wang, S., Wan, J., Zhang, D., Li, D., and Zhang, C., (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101, pp. 158-168.

Weiss, G., (1999). Multiagent systems: a modern approach to distributed artificial intelligence. MIT press.

Wooldridge, M., (1997). Agent-based software engineering. *IEE Proceedings-software*, 144(1), pp. 26-37.

Wooldridge, M., (2009). An introduction to multiagent systems. John Wiley & Sons.

Wooldridge, M. J., and Jennings, N. R., (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2), pp. 115-152.