This is an Accepted Manuscript of an article published by Taylor &Francis in International Journal Of Computer Integrated Manufacturing
https://doi.org/10.1080/0951192X.2021.2022760

ARTICLE TEMPLATE

Knowledge representation in Industry 4.0 Scheduling problems

Daniel A. Rossit[a] and Fernando Tohmé[b]

[a]Departamento de Ingeniería, Universidad Nacional del Sur and INMABB-CONICET, Bahía Blanca, Argentina; [b]Departamento de Economía, Universidad Nacional del Sur and INMABB-CONICET, Bahía Blanca, Argentina

ARTICLE HISTORY
Compiled March 22, 2024

ABSTRACT
Industry 4.0 raises the need for a closer integration of management systems in manufacturing companies. Such process is driven by Cyber-Physical Systems (CPS) and the Internet of Things (IoT). Starting from the potential of these technologies, a knowledge architecture aimed at addressing scheduling problems is proposed. Scheduling-support systems generally do not solve real-world scheduling problems, being instead only capable of solving simplified versions, producing solutions that human schedulers adapt to real problems. The architecture aims to record and consolidate the empirical knowledge generated by the solutions of actual scheduling problems. In this way, it summarizes the implicit criteria used by human schedulers. The architecture presented here records this knowledge in data structures compatible with the structure of scheduling problems. In further iterations this knowledge crystallizes into a sound and smart structure.

KEYWORDS
Cyber-Physical Systems; Industry 4.0; Scheduling; Decisional DNA; Knowledge representation

1. Introduction

The fourth industrial revolution, known as Industry 4.0 (Hermann, Pentek, and Otto 2016) aims, like its predecessors, to change the way in which manufacturing processes are carried out (Zhang et al. 2020). Industry 4.0 has already started to show that the capacity and flexibility of production processes in industrial environments at the technological frontier have greatly improved (Zhang et al. 2018). This enables the creation of new business models based on the customization of manufactured products, a trend that will continue generating economic niches in the near future (Yu et al. 2017).

As with past industrial revolutions, the paradigm shift is driven by technological advances that modify the structure of production systems (Zhong et al. 2017). In the case of Industry 4.0, the main technological advances behind the new production structures are *Cyber Physical Systems* (CPS) and the *Internet of Things* (IoT) (Lee, Bagheri, and Kao 2015; Monostori 2014). CPS are production systems that allow the direct integration of the physical space in which production takes place with the cybernetic or digital space usually associated with decision-making processes (Wang, Zhang, and

CONTACT Daniel A. Rossit. Email: daniel.rossit@uns.edu.ar

Zhong 2020). Basically, CPS are systems that consist of physical components with computational functionalities (Guo et al. 2020). This is why CPS allow to create a real-time model of the physical processes in cyber space, yielding what is known as a *Digital Twin* of the real world counterpart (Park et al. 2019; Zhang et al. 2020). The different CPS connected by IoT can collect data and communicate with each other, improving the modeling capacity of Digital Twins (Leng et al. 2019). In addition, IoT transmits the same information in real time to decision-making centers, increasing the capacity of controlling online the production process (Tao et al. 2018). The impact of these technologies makes it possible to conceive that in the near future, production processes will be entirely based on CPS (Monostori 2014). Their use ends up resulting in the digitization of the production process, availing the use of increasing amounts of data and information, improving the decision-making processes in industrial firms (Kusiak 2017; Zhang et al. 2018; Rossit, Tohmé, and Frutos 2019a).

Scheduling is one of the decision processes that has been most affected by the advent of Industry 4.0 as well as by the widespread use of big databases (Ivanov et al. 2016; Rossit and Tohmé 2018). Scheduling problems arise at the last stage of production planning, involving the final decision-making phase before starting the actual production activities (Pinedo 2012). Scheduling takes care of assigning the different work orders to the production resources respecting established time horizons (Framinan, Leisten, and García 2014). This poses problems that are very difficult to solve computationally (Lenstra, Kan, and Brucker 1977), making scheduling a nontrivial activity (Framinan and Ruiz 2010; Qin et al. 2019). The scientific community has only very recently started to study how to make scheduling an inherent task of Industry 4.0 environments (see, for instance (Ivanov et al. 2018; Dolgui et al. 2019; Rossit, Tohmé, and Frutos 2019d)). Currently, human schedulers use *Decision Support Systems* based on *Advanced Scheduling Planning* modules as those offered by SAP or SIEMENS' Preactor (Bożek and Wysocki 2015). These systems are usually integrated into *Manufacturing Scheduling Systems* (MSS), which manage and execute the production itself (Kletti 2007). The tools used to find solutions to scheduling problems require a very intense participation of the human scheduler, who has to evaluate potential alternatives using Decision Support Systems (Leusin et al. 2018). Furthermore, the solutions so obtained cannot be adapted straight ahead to the real problem at hand (Ferraro et al. 2019). The scheduler has to adapt manually those solutions to the real problems. This implies that the actual solutions adopted depend heavily on the scheduler in charge. Imponderable factors threaten the quality and efficiency of a plan, on top of the inefficiencies already present in solving manually such a complex problem (NP-hard in most cases) (Framinan, Leisten, and García 2014).

Based on the potential of Industry 4.0, it is presented a design intended to overcome these difficulties, automatizing as much as possible the process of scheduling. The innovation is to incorporate an Artificial Intelligence (AI) complement to the MSS based on SOEKs (*Set of Experience Knowledge Structure*) and *Decisional DNA* (DDNA) (Sanin and Szczerbicki 2007; Shafiq, Sanín, and Szczerbicki 2014). These two latter concepts involve a tailored knowledge representation system which allows the generation of an efficient repository of decision-making events (Sanin et al. 2012). Therefore, the AI complement added to the MSS aims to efficiently register the empirical knowledge generated by schedulers. The AI complement records the adjustments to the MSS solution found by the schedulers. This generates a repository of formal decision-making events modeled through data structures (SOEKs) that are compatible with scheduling problems and their solutions. A search engine appended to the system matches new scheduling problems to formal decisions made for similar instances, already recorded in the repository. Therefore, the response capacity and quality of the system improves in time.

The main contributions of this paper are:

- A proposal of an architecture able to record the solutions that a human planner generates in time that will be later provide the grounds to analyze unforeseen events.
- The design of the architecture will support the ability to describe in detail the scenarios at which the decisions made by the scheduler were generated.
- A contribution to the design of expert systems for the solution of scheduling problems.

The rest of this work is organized as follows. In Section 2 the main concepts used of the design are introduced. Section 3 presents the decision-making process involved in Scheduling and defines the problem that is intended to address. Section 4 presents the proposed architecture and discusses its main features. Section 5 presents the conclusions of this work.

2. Industry 4.0 and knowledge representation

In this section it is presented the characterization of the industrial environments under the Industry 4.0 paradigm and the main ideas behind the proposed design. The focus is centred on the application of ideas drawn from Knowledge Representation in Artificial Intelligence that can be useful for the mentioned purposes, in particular on the concepts of SOEK and DDNA , which allow modeling different decision-making problems.

2.1. *Background on Industry 4.0*

Industry 4.0, unlike past industrial models in which the management proceeded through hierarchical and centralized structures, presents schemes in which autonomous agents interact in decentralized architectures. These agents are connected to each other and to the decision centers through IoT (Tao et al. 2018). These connections make it possible to greatly improve the flow of data previously registered at the PLC (Programmable logic controller) or SCADA (Supervisory Control And Data Acquisition) level, which can now be transmitted to the other stages of the production process as well as to the decision-making centers (Rossit, Tohmé, and Frutos 2019c). This increase in the ability to transmit data necessarily implies a considerable increase in the amount of data circulating within an Industry 4.0 environment (Tao et al. 2018; Qian et al. 2020). This is why, among the technologies that drive the paradigm shift in Industry 4.0, data science is vitally important (Zhong et al. 2017). These technologies allow to handle and manage large amounts of data efficiently and achieve a more realistic representation of the production system. In turn, the CPS are linked through IoT, becoming able to access tools that allow to easily increase computer processing capacity. This can be achieved through Cloud Computing or Edge Computing, which allows a faster response (Tao et al. 2018). Processing this large amount of data requires the techniques already developed for large non-homogeneous databases collected online (Kusiak 2017).

2.2. *Knowledge representation*

In the era of mass digitalization, data is generated in large amounts by all systems and in all environments (Liu et al. 2019). Production processes are not an exception, generating more than 1000 exabytes per year (Tao et al. 2018). The data comes from different areas of the industry, such as the shop floor, the negotiations with suppliers, the results of medium and long-term planning, and many other sources. Such a variety of sources means that the data is stored in various formats, in such way that no person or group of people can understand and use them all (Wang, Sanín, and Szczerbicki 2013). On the other hand, new market trends indicate that players who can take advantage of the data by transforming it into useful information that enjoys large competitive advantages in the market (Yu et al. 2017; Jung 2009). Therefore, new solutions and techniques are needed to extract knowledge from the available data that, very often, can be unstructured, semi-structured, diffuse or vague. In addition, it is important to ensure the transportability, reusability and the ability to share the knowledge extracted from that data (Jung 2009).

One of the main issues to address in order to convert data into useful information is the proliferation of different knowledge representations, and consequently of different knowledge management systems (Alavi and Leidner 2001). Finding a unifying representation of knowledge is a complex, hard to accomplish, task. Thus, in this work the main focus is to consider the pros and cons of two commonplace conceptions. One assumes a hierarchical view while the other is much more flexible.

The hierarchical vision of knowledge was introduced by Rowley (Rowley 2007), stating that addressing and interpreting reality requires a hierarchical procedure organized according to the Data-Information-Knowledge-Wisdom structure, commonly called the DIKW hierarchy (Figure 1). At the lowest level there is data, which is not, by itself, useful to the decision maker. Data has no meaning or value without a context or interpretation. Data, thus, provides only elementary descriptions of things, events, etc. Information, instead, can be identified with a dataset organized in such a way that it becomes able to convey meaning to the decision maker. This means that in this view information is at a higher level than data. According to Rowley (Rowley 2007), "knowledge is an intrinsically ambiguous and equivocal term". For the purposes of this work, knowledge can be understood as a synthesis of various sources of information over time, where the structures of beliefs, experience and ability of the entity that builds knowledge influences the process. Knowledge can be seen as a combination of information, understanding, ability, experience, skills and values. The highest term in the hierarchy, wisdom, has an even more abstract meaning, which can be roughly assimilated to accumulated knowledge, which allows the entity to understand how to apply concepts of a domain to new situations or problems (Jessup and Valacich 2003). Figure 1 shows the relationships among the objects in this hierarchy. In stark contrast, the ability of computer systems to process these items runs in the opposite direction: the higher an element is in the hierarchy, the harder becomes to process it algorithmically.

A perspective closer to business, presented by Alavi & Leinder (Alavi and Leidner 2001), suggests that it is not possible to define a strict hierarchy like the DIKW model. These authors reviewed the literature on Knowledge Management at their time, concluding that hierarchical definitions of knowledge (data-information-knowledge) always depend, at some point, on some arbitrary choices. They adopt the definition in (Fahey and Prusak 1998), suggesting that there is no knowledge independently of a cognitive agent, being shaped both by her needs and her initial stock of knowledge.

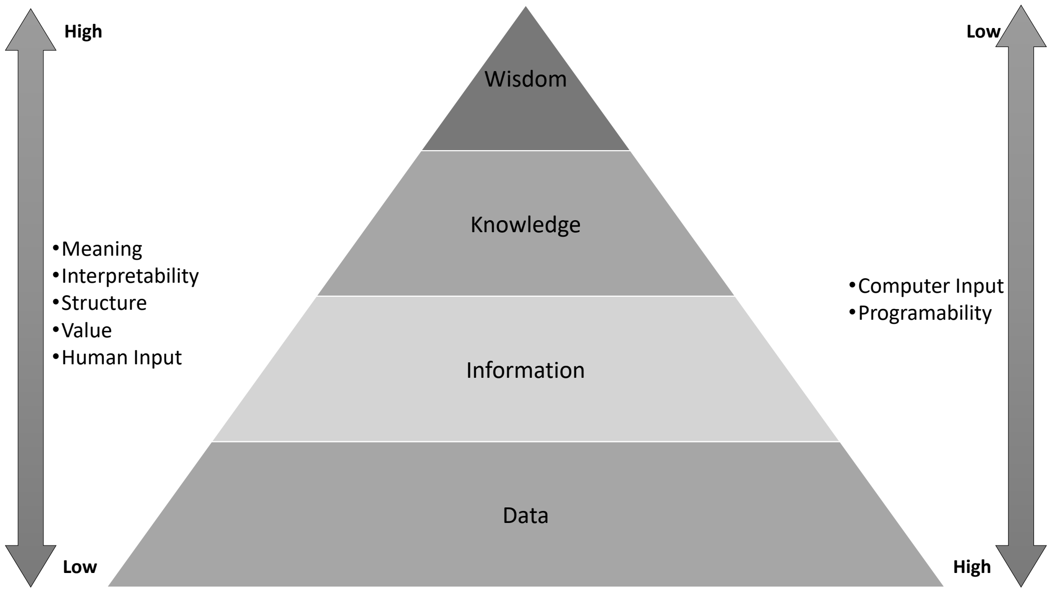

Figure 1. The DIKW hierarchy (Rowley 2007).

Therefore, Alavi & Leinder state that there cannot exist a radical difference between information and knowledge other than by the fact that information becomes knowledge once it is articulated in the minds of individuals. Conversely, knowledge becomes information once it is articulated and presented in the form of text, graphics, words or other symbolic forms. This approach is consistent with the classical definition of knowledge of Churchman (Churchman 1971), which indicates that knowledge is defined by the user and not by any underlying collection of data. According to this, the definition given by Alavi & Leinder is: *knowledge is a justified belief that increases an entity's capacity for taking effective action.* They also postulate two important general characteristics: (i) since knowledge is personalized, to make it useful to others, it must be expressed and communicated in a way that is interpretable by the recipients, and (ii) large volumes of information are only valuable (in particular Enterprise Knowledge Management Systems) if they are useful for its recipients. Therefore, knowledge can be hard to transfer through an organization, even though IT technologies make it easily accessible.

The authors of this work remain agnostic with respect to these two opposed views and concentrate on a practical approach in which knowledge becomes "a significant high-level tool that allows the decision maker to improve her decision process". Accordingly, the following structures representing decision-making events to exploit them using methods drawn from data science are presented.

2.3. *Decisional DNA*

Decisional DNA is a Knowledge Engineering technology aimed at solving decision-making problems. Usually, when a decision event arises, managers select actions that have previously worked well. They detect the most significant features of the current circumstances that allow the identification with similar situations and thus apply the corresponding actions that gave good results in the past. Therefore, it is very important to keep a record of previous decisions and turn them into explicit knowledge (Sanin and

Szczerbicki 2007). Decisional DNA (DDNA) stores previous Formal Decision Events (FDE) explicitly, generating a Set of Knowledge Experience (SOEK) (Sanin et al. 2012).

SOEK is a formal model of decision-making knowledge based on real world evidence. It starts by classifying the components of decision-making events as being either Variables, Functions, Constraints or Rules (Sanin et al. 2012). Variables constitute the core of SOEKs, expressing the states of events. Functions are formulated as equations intended to describe the relations between dependent variables and the set of input variables. A decision may differ from another by the addition or subtraction of a function. Constraints are similar to functions in that they relate variables but their purpose is different, since they seek to restrict the performance and configuration of the system as well as the feasible solutions to the decision problem. Finally, rules provide yet another way of relating variables, conditioning the relations among variables, essentially using the IF-THEN-ELSE format to connect preconditions and their consequences (Wang, Sanín, and Szczerbicki 2013).

These components are stored in a dynamical combined structure inside a SOEK. This is analogous to the way in which four nucleotides are combined in DNA, giving a distinctive character to the result (Sanin et al. 2012). Besides, the elements in the structure are connected among them, imitating a DNA chain, i.e. a gene. Therefore, a gene can be assimilated to a SOEK: in the same way as a gene produces a phenotype, a SOEK produces a decision value in terms of its elements. Such value is called the *efficiency* or the *phenotype value* of the SOEK (Shafiq, Sanín, and Szczerbicki 2014), being the response to a query. Analogously to the way in which a gene guides the hereditary responses of a living organism, a SOEK leads to responses in certain areas of a decision process.

A single SOEK cannot control a complete system, not even an area or specific category of a system. And therefore it is necessary to acquire and build larger sets of experiences. The regular operations of a system yield a large number of decisions and thus a collection of different SOEKs (Shafiq, Sanín, and Szczerbicki 2014). A group of SOEKs of the same category constitute a decisional chromosome in the same way as DNA does with genes. This decisional chromosome stores the decision "strategies" of a category. In such case, each module of chromosomes constitutes an entire inferential tool, providing a schematic view of knowledge in an organization. More precisely, a diverse set of decisional chromosomes provides a family of inferential strategies in an industrial organization (Sanin et al. 2012).

3. Production Scheduling decision-processes and Industry 4.0

This section presents a solution to the problem of making decisions on Production Scheduling. For this, Manufacturing Execution Systems (MES) are analyzed from the perspective of Industry 4.0 technologies.

3.1. *Manufacturing Execution Systems and Industry 4.0*

Manufacturing Execution Systems (MES) are in charge of controlling, executing and managing all the actions that are directly related to production processes. Taking the ANSI /ISA-95 architecture as a reference framework, MES are at level 3, below level 4 (general management and ERP systems), and above level 2 (SCADA and control systems of the production). The typical function of a MES is the execution of

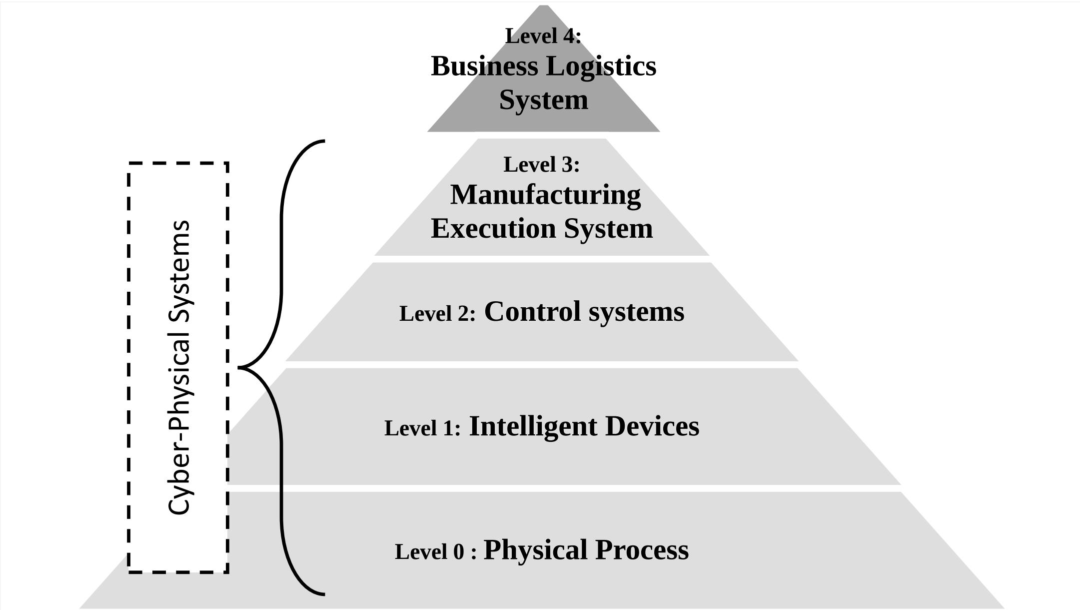

Figure 2. Levels of ANSI/ISA-95 absorbed by CPS.

production plans defined by the organization at Level 4. MES provides information that helps decision makers to understand how current plant conditions can be optimized to improve production (Kletti 2007). Although MES implementations depend on the type of production (batch, continuous or discrete) and on how firms implement it (especially for SMES), for the VDI (Verein Deutscher Ingenieure), the MES must provide, among other functions, detailed planning and detailed scheduling control, operating resources management, material management, personnel management, data acquisition and processing, performance analysis and quality analysis.

Industry 4.0 has a natural impact on MES and the problems and scenarios in which MES provide support. This impact is essentially due to CPS as they allow a direct link between virtual and physical spaces (Lee, Bagheri, and Kao 2015). A MES usually processes the data and information from different sources (production plans, stock of materials, production quality control and shop floor status, among others), and accordingly generates reports or computes schedules that lead to achieving the production objectives of the firm. The decision maker analyzes the output of the MES and generates a plan (Kletti 2007). The CPS shorten the distance between the space where the MES work (virtual-computational) and the physical production one, allowing the CPS themselves to reconfigure the shop floor autonomously and efficiently (Monostori 2014).

To illustrate the impact of the CPS on the production systems it is depicted in Figure 2 the hierarchical structure ISA-95 and the levels reached by the CPS. It can be seen that CPS included all the management systems of the industry from level 0 (physical process) to level 3 (MES) (Rossit, Tohmé, and Frutos 2019b; Rossit and Tohmé 2018). This is due to the ability of CPS to carry out a broad spectrum of activities, ranging from physical production operations (level 0) to planning, evaluation and management of the entire production process (level 3), through control of actions and systems at levels 1 and 2 (i.e. measuring and detection instruments, as well as control systems). Some of the direct benefits of this integration of functionalities are, for example, the greater flexibility to respond to unexpected events or the faster transmission of information throughout the entire system. These advantages are due to the fact that CPS can translate the data obtained at level 1 into the higher order language used at level 3, bypassing the adjacency limitations inherent in ISA-95, generating faster responses to unforeseen events.

It can also be seen in Figure 2 that decisions at a higher hierarchical level are outside the control of the CPS. Basically, it is represented the fact that, although all the flexibility and information provided by the CPS allow to improve hierarchical decision making, these decisions will still be made by human beings. Decisions at the aggregate level (such as company objectives) will be handled by ERP and human systems, already adapted to intelligent manufacturing environments. A relevant detail is that the CPS manage a good deal of the decisions made by the ERP systems (such as inventory control, database management, information management on suppliers, etc.), but do not absorb them entirely (Rossit, Tohmé, and Frutos 2019d).

3.2. *Scheduling problems*

To solve a scheduling problem, the allocation of available production resources to a production plan generated at a previous planning stage must be resolved. A detailed description of the process is necessary to define the schedule, which implies the handling of a large volume of data and conditions (Framinan, Leisten, and García 2014; Rossit, Tohmé, and Frutos 2019d). As is intuitively evident, these decision problems have a strong combinatorial nature and, consequently, a high complexity.

Formally, a scheduling problem consists in the allocation of a family N of jobs, $N = \{1, 2, \ldots, n\}$ on a set M of machines, $M = \{1, 2, \ldots, m\}$. Each job j consists of a class O_j of operations, where operation O_{ij} of job j must be carried out on machine i. Each operation O_{ij} has an associated processing time p_{ij} on machine i. Each job j will be associated to an ordering R_j of the operations of O_j, reflecting the precedence ordering among operations. The whole point of scheduling is to find a schedule π of jobs over machines yielding an optimal value $F(\pi)$, where F denotes some objective function.

Scheduling problems are highly dependent on the actual details of the production setting (Job Shop, Flow Shop, etc.). This implies that different parameters (delivery dates, preparation times, waiting times, etc.) and different objective functions (makespan, total tardiness, maximal tardiness, etc.) require alternative statements of the general problem.

Given the combinatorial nature and complexity of most scheduling problems, Decision Support Systems are required to assist in the corresponding decision making process (Framinan, Leisten, and García 2014). The resulting systems are known as Management Scheduling Systems (MSS). To delve into how to design an ad-hoc MSS system, Framiñán & Ruiz (Framinan and Ruiz 2010) presents a guide for the design, implementation and testing of an MSS. A classical MSS model was proposed by Pinedo (Pinedo 2012), who presented a clear overview of MSS 3.

The system consists of the following components: a Database Management module, an Automatic Schedule Generator, a Schedule Editor and a Performance Evaluator. The user accesses the last two modules through a graphical interface (GUI), while the Database Management module manages all the information required to develop a schedule (Rossit, Tohmé, and Frutos 2019a). The Data Management module takes as inputs the schedules that must be fulfilled, that is, the production orders and the master production programs, but also receives inputs from the plant data, which

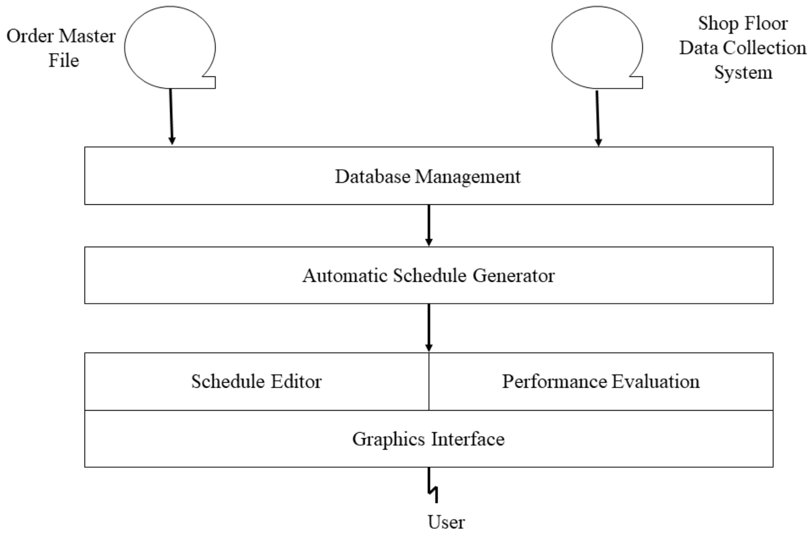

Figure 3. Standard Scheduling System (Pinedo 2012).

allows monitoring the status of the physical aspects of the production. The output of the database management module feeds the Automatic Schedule Generator.

The MSS, represented in 3, is intended as an auxiliary or support system for decision making by the programmer or the end user. The purpose of the MSS is to achieve a work schedule of the entering production orders and also to address the events that arise in the dynamics of the production process (Pinedo 2012). The main tasks faced by system users are the assignation of jobs to resources (in general, machines), handling problems that affect programs (such as changes in resources, dates, quantities, etc.) and anticipating problems futures with the program (Framinan, Leisten, and García 2014; Lv, Zhang, and Qin 2019). In a field study, McKay & Buzacott (McKay and Buzacott 2000) identified that in order to fulfill these tasks, planners usually follow a similar process or "script", either explicitly or implicitly, independently of the specific production area. In the case of schedulers, this process begins by evaluating the current production situation in search of possible "crises" or sources of conflict, as for instance when a job is being processed taking a longer time than planned or by misusing materials. Once the focus of the crisis has been identified, the scheduler analyzes whether a reschedule or a reallocation of resources should be generated. Then, they update the scenario and continue with the process of evaluation and analysis of possible future crises, in a recursive way.

A consequence of this type of management (MSS managed by different people), is that the behavior of the system becomes affected by the characteristics of the actions of the individuals who make scheduling decisions. One of the characteristics detected in field studies is that planners usually have a "myopic" perspective of the situation, not being able to see more than half an hour ahead (Framinan, Leisten, and García 2014). This myopia originates from two sources: on the one hand, the high complexity of Scheduling problems, and on the other, the changing scenarios in which these problems arise. Both conditions make it hard for a scheduler to contemplate a farther horizon when looking for critical breaks.

On the other hand, planners manage to reduce the size of problems quickly, applying criteria such as Drum-Buffer-Rope and focusing the effort on bottlenecks (Framinan, Leisten, and García 2014). This condition itself can be both an advantage and a disadvantage. On one hand, reducing the size of a problem drastically facilitates the resolution process. However, an excessive reduction of the problem can eliminate optimal solutions, or at least valuable ones if quasi-optimal methods are used. Another aspect of the behavior of schedulers is the variability of their recommendations, since two schedulers assigned to different shifts can solve the same situation quite differently (Framinan, Leisten, and García 2014). A related aspect is that planners usually design schedules guided by the objectives they seek to optimize. This means that, instead of generating a Schedule following some standard procedure or method, they evaluate modifications by the potential objective satisfaction to obtained thanks to those modifications. In this sense, the field studies of Vernon (Vernon 2001) indicate that in such cases the planners seek to improve their own production objectives instead of objectives related to the service level of the shop floor, which may lead to outcomes that do not respond to the expectations of management.

3.3. *Dynamic and Standard Manufacturing Scheduling*

A Manufacturing Scheduling System (MSS) consists of a set of business functions that the system controls in the context of supporting production management decisions. With respect to this, Framiñán & Ruiz (Framinan and Ruiz 2010) state that these functions can be in general classified in two classes, depending on the time horizon. These classes address, respectively, a global and a local temporal level:

- A higher level that uses the output of production planning to set up the dates for the beginning of each job on each machine. The activities at this level are often referred as release scheduling.
- A lower level which is involved with real-time item movement planning. The actions at this level are usually called reactive scheduling.

An MSS must cover these two levels adequately, that is, the system architecture must have functionalities capable of monitoring and executing the planned schedules (Wang et al. 2019). Mckay & Wiers (McKay and Wiers 1999) proposed the concept of “sustained control”, which implies that planners have to be able to monitor the progress of production and solve problems if production deviates from the original plan. Faced with these situations, schedulers may not have to solve scheduling problems by themselves, but some particular optimization problems included in scheduling decision processes. Therefore, the MSS must support both levels. The user tends to intervene more frequently at the reactive scheduling level.

The reactive scheduling problem arises once the schedule is generated and manufacturing operations begin . Managers and supervisors want the shop floor to run according to the schedule (Qin, Zhang, and Song 2018). In practice, operations tend to deviate from the scheduled plan. Small deviations in the starting and ending times are to be expected, which are generally ignored (the definition of small depends on the system in question). Larger deviations occur when unexpected events interrupt the initial schedule (machines fail, for instance). Even if managers and supervisors do not explicitly update the schedule, modifications are introduced when schedulers react to interruptions that delay the completion of tasks or carry those out in a disorderly manner (Vieira, Herrmann, and Lin 2003).

Rescheduling is the process of updating an existing production schedule in response to interruptions or other changes (Ouelhadj and Petrovic 2009). Some of the factors that usually trigger these reschedules are the arrival of new jobs, failures or repairs of machines, changes in delivery dates, delays or shortages in the delivery of materials, changes in the priorities of work, reprocessing or quality problems, over or underestimation of processing times, absenteeism of operators, etc. To solve these problems, the planners use different techniques or strategies, ranging from addressing the problem in a completely reactive way to generate robust strategies giving the initial schedule enough slack to "absorb" these events (Framinan, Leisten, and García 2014; Vieira, Herrmann, and Lin 2003; Ouelhadj and Petrovic 2009; Rossit, Tohmé, and Frutos 2019c).

3.4. *The problem in an Industry 4.0 perspective*

MSS generally calculate solutions for the release scheduling problem with some deficiencies. These are mainly due to the fact that real industrial environments present too many restrictions as to even conceive to seek a real optimal solution (Framinan and Ruiz 2010). In practice, simplifications are usually considered, either in the modeling of the problem or in the method of resolution. When the problem is simplified, the optimization is carried out using some method (for example, ad-hoc designed metaheuristics) that may work well for the simplified version. For example consider a flexible hybrid flow shop system with sequence-dependent setup times, but omitting personnel selection, time windows, machine life before maintenance shutdown, etc. In that case, a very good solution for the simplified problem can be obtained. Then, the scheduler can manually modify the solution as to approximately meet all the real restrictions of the problem (Framinan and Ruiz 2010). Alternatively, a simple resolution approach can be used (for example dispactching rules) allowing fast solutions even for problems with too many restrictions (Framinan, Leisten, and García 2014; Pinedo 2012). However, this solution can be very poor, and the scheduler may be forced to modify the solution (at least partially) to achieve a better performance. For reactive scheduling problems, the question of rescheduling upon a disruption is answered in a similar way. When adjusting the computed solution to the real problem, the extra conditions generated by a reschedule should be considered, like new delivery dates, other than material handling, different windows for usable times, etcetera (Vieira, Herrmann, and Lin 2003).

Given this, it remains to see whether Industry 4.0 can help to improve the outcome. With regards to the limitations of modeling the problem (solving a simplified version) or the limitations in computing solutions using simplistic resolution approaches, there is not much to be achieved by the application of Industry 4.0 technologies, or at least not directly, since they do not increase the hardware capacity or the modeling techniques used to solve those problems. Nevertheless, Industry 4.0 allows the incorporation of Artificial Intelligence and Data Science tools that can contribute to improve the way in which problems are addressed. Specifically, by generating a system trained by the schedulers, collecting the "adjustments" made by them in problems of release and reactive scheduling (either in modeling or solving the problems). That is, Industry 4.0 allows to create a system that collects data and information in an intelligent and efficient way, so that when a scheduler faces situations similar to those of the past she can count on a support system to provide at least the best of the empirical knowledge generated by her colleagues. This leads to speed ups in the decision process, increasing the response capacity and laying the groundwork for improvements in the solutions offered by the support system.

4. Decisional DNA approach to scheduling

To solve the problem discussed in the previous section, incorporating an Artificial Intelligence complement to the MSS is proposed. Then, it is introduced a design for a complement based on DDNA, which can be associated to the scheduling engine of the system. Then, the whole architecture is presented, where both the scheduling classic complement and the DDNA complement are linked, evaluating its potentialities.

4.1. *The DDNA Complement*

The idea behind this architecture is that it is convenient to add an AI complement to a MSS, able to translate human reasoning processes into usable and understandable procedures for CPS. This complement could be designed on the basis of the Decisional DNA technology. As a proof of concept it can be shown that each element in the decision-making process that is not usually considered in an optimization process (i.e. solving job sequencing) can be modeled in this way. Those elements are usually incorporated by the user at the solution-tuning stage. The DDNA complement proposed here, can collect those adjustments and stored them for similar events in the future.

To embed this decision process into a CPS it is necessary to model it in the adequate terms for an intelligent system. In this sense, the concept of SOEK (Sanin and Szczerbicki 2007), presented before in this paper, gains relevance. Basically, it translates a formal decision event (FDE) into a model based on variables, functions, constraints and rules. The analogies of this model with standard optimal scheduling problems (Pinedo 2012) are quite evident. More precisely, scheduling problems can be formulated in almost the same terms as a SOEK, albeit the Decisional DNA's independent variables (which do not get modified in the scheduling process, as for instance processing times, precedence of operations, etc.) are called parameters. Then, the FDE recorded by SOEKs, can be stored by the standard information of the optimization process, jointly with some other elements, like environmental features or un-planned events. This idea is illustrated in Figure 4, where the User decision-making process is influenced by the standard scheduling information, and also, by the un-planned events or environment aspects. But, the most important point is that all that information has an associated performance measure, which can improve future decision-making in similar situations. That is, the SOEK can record the shop-floor condition, the solution adopted and the performance or actual result of that solution.

A SOEKS system consists of four interacting modules. The first one is the Integrator, which compiles and organizes the information of the scenario under study. In this case it will associate each decision-making experience with a description of the context in which it happened, its nature, its consequences, etc. The Integrator sends this information to the Prognoser, which analyzes it. In interaction with the DDNA Repository and the XML Parser, the Prognoser organizes the knowledge required by the task at hand. The Convertor translates this knowledge into DDNA and stores the results in the Repository. The latter is the core of this design, since it is in charge of keeping track of previous experiences while also influencing the future generation of chromosomes. The entire system links with an I/O subsystem that informs in real time the DDNA of all the relevant external events. The latter, in turn, evaluates each of those events and checks whether there exist previous solutions to similar problems.

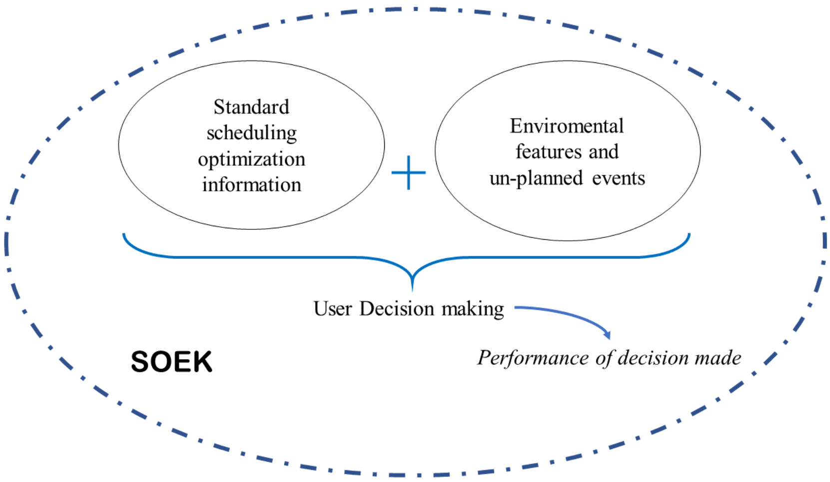

Figure 4. Information registered in a SOEK.

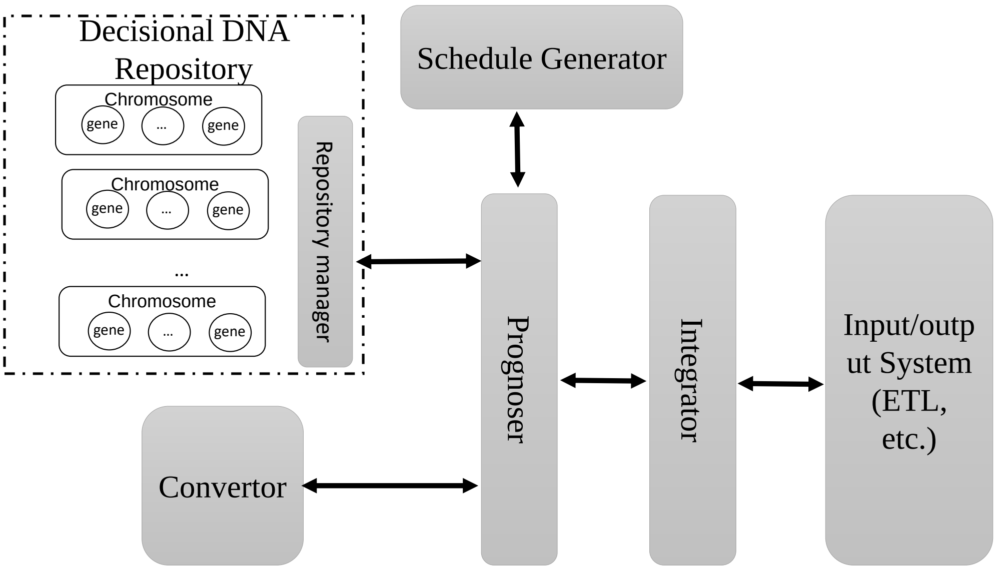

Figure 5. System Design for DDNA Scheduling System.

If so, it connects the event to those solutions and triggers the corresponding actions stored in the DDNA Repository.

In order to ensure that a DDNA system has these abilities, it must be trained, internalizing the knowledge needed to gain them. This DDNA complement to the Scheduling Enhanced CPS, can obtain this knowledge in two different ways: 1) by working in parallel with the human scheduler on the normal day-to-day operations, 2) by associating it to the scheduling generation process. In the first case the DDNA system will collect the FDE solved by the scheduler, generating the corresponding DNA. Since the production system is commanded by CPS, even the real-time modifications of the schedule (not recorded by the human scheduler) can be collected. The DDNA system will get information about them through an ETL system (Extract, Transformation and Load) already in the network of the CPS. Thus, if a machine runs an operation or job that was not originally scheduled, the CPS network will record this event and, through the ETL system, the DDNA will generate knowledge about it. Alternatively, the DDNA can access a database of past FDE, from which it generates knowledge.

Another way of generating knowledge is required when the DDNA system is unable to respond to a given FDE since the DDNA repository does not have a similar record. But then, the DDNA is able to translate such event or scenario as a SOEK that, as said, is equivalent to a classical scheduling problem. This problem is thus submitted to the Schedule Generator of the MSS. In summary, the DDNA system is, potentially, able to address any scheduling problem since any action executed ends up being a solution to a scheduling problem.

4.2. *Assessment of the design*

The proposal of incorporating a complement of knowledge representation based on SOEKs and DDNA to the MSS, is depicted on Figure 6. This figure shows the relationship between the MSS and DDNA complement at the user level. The DDNA complement is in charge of monitoring the status of the shop-floor. Then, DDNA can function as a library of past solutions associated to different scenarios. The overall system leverages the ability to make decisions of the scheduler. Thus, the architecture depicted in Figure 6 can be associated to different shop-floor configurations, such as cell manufacturing, assembly lines, job shop, flow shop, etc. At each case, the type of scenario could be different and the MSS should be properly adopted.

Despite this versatility to adopt different shop-floors configurations, it remains necessary to discuss whether this architecture can handle the two levels of decision-making described in Section 3.3, namely, release scheduling and reactive scheduling (on-line). The proposed architecture contributes, indeed, to the resolution of scheduling problems at its two levels. With respect to the problem of release scheduling, this architecture contributes providing better starting points, i.e. potential solutions. This criterion is proven in practice, since it is often useful to solve scheduling problems by means of backtracking algorithms that seek more recent solutions to the problem (Pinedo 2012) . By backtracking, the MSS and the scheduler review the recent history of schedules. Using, instead, a DDNA-based add-on the history of past schedules and scenarios increases the chances of finding quickly a good solution that can be used as raw material for the process of generating a new release schedule. In addition, DDNA incorporates more information about the kinds of scenarios associated with each FDE, accumulating a curated history of recent resolutions. This has a non-trivial consequence, since backtracking may recover *simplified* problems corresponding to different real problems while DDNA increases the possibility of matching the current problem to an *actual* problem solved in the past.

With respect to the reactive scheduling problem, the proposed architecture can show how a previous solution was obtained and the results that ensued from its application. Therefore, for a current event, the scheduler can either improve a previous good schedule or even use the very same solution, speeding up the response time.

In turn, DDNA homogenizes the response capacity of all schedulers, that is, of the different planners that work in different shifts. They all gain access to an expert system yielding solutions according to the best criteria (maybe generated and executed by another scheduler). In this way, all the schedulers will always get the best out of past cases.

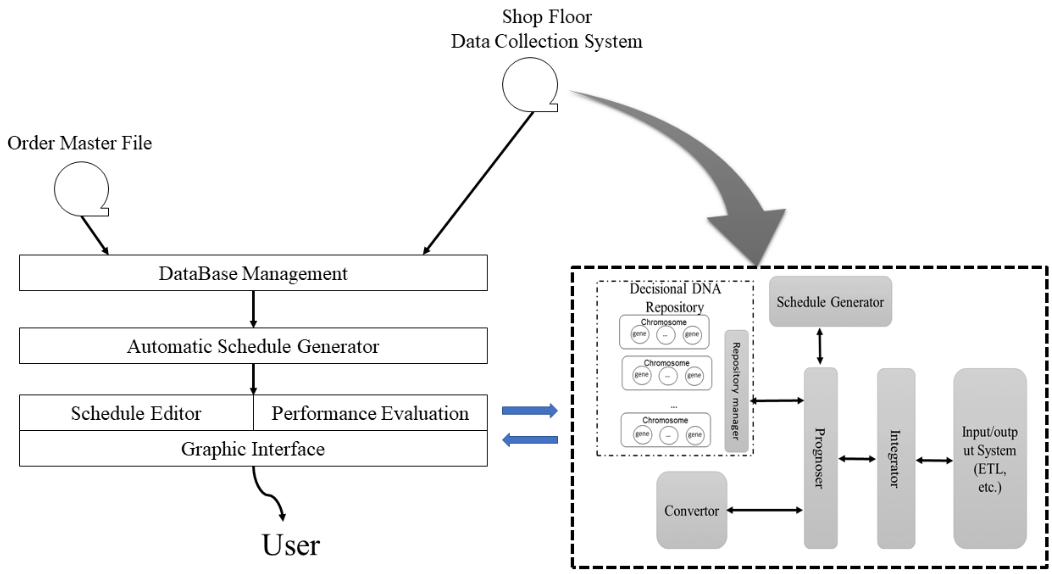

Figure 6. Complete System Architecture for Scheduling System with MSS and DDNA complements.

4.3. *Case Study: Single Machine Scheduling problem*

In order to show how the proposed architecture may work on scheduling problems, the following case study is presented.

Assume a single-machine process in which the goal is the minimization of total weighted Tardiness. As a particular feature consider sequence dependent setup times. While simple, this setting is enough to illustrate the knowledge representation architecture proposed in this paper. The description of this problem in a three-field notation $(\alpha \mid \beta \mid \gamma)$ is $1 \mid s_{jk} \mid \sum w_j T_j$. The tardiness of job j, T_j , can be calculated in terms of j's due date, d_j, and j's date of final processing, c_j, as: $T_j = max\{c_j - d_j, 0\}$. The weight w_j represents the relative value of job j with respect to the other jobs. The setup times s_{jk} depend on the sequence, and thus, it is not equivalent to start processing k after finishing j than the other way around. The goal is to minimize $\sum w_j T_j$ respecting the constraints of the problem.

This single-machine problem is NP-Hard when the weights of the jobs are all equal (Du and Leung 1990), while it is Strong NP-hard if they are different (Lawler, Lenstra, and Kan 1982). Different heurıstic and meta-heuristic methods have been proposed for this problem (Tasgetiren, Pan, and Liang 2009; Bektur and Saraç 2019). Also dispatching rules have been suggested for treating it (Park, Kim, and Lee 2000; Pfund et al. 2008). An effective dispatching rule is the Apparent Tardiness Cost with setups (ATCS rule) (Lee and Pinedo 1997). This rule associates two different dispatching rules, the *Weighted Shortest Processing Time first* ($WSPT$) rule (which orders the jobs according to the lowest weighted processing times), and the *Minimum Slack first* (MS) rule (which takes into account the slack time of each job, determining $max\{d_j - p_j - t, 0\}$, being t the moment at which slack is evaluated). The ATCS computes the time at which a machine gets free, and orders the jobs according to decreasing priority defined by the index. This rule was presented first in an exponential version in (Vepsalainen and Morton 1987)), while in (Lee and Pinedo 1997) it was extended to the case of sequence-dependent setup times, in which the index is defined as:

$$I_j(t,l) = \frac{w_j}{p_j} exp(-\frac{max(d_j - p_j - t, 0)}{K_1 \bar{p}}) exp(-\frac{s_{lj}}{K_2 \bar{s}}) \quad (1)$$

where K_1 and K_2 are scale parameters for the due date and the setup functions of job j, respectively. These parameters strongly depend on the specific problem under analysis and must be defined by the scheduler (Xi and Jang 2012). They are, in general, associated to empirical statistics calculated for the problem: the due date tightness factor (τ), the due date range factor (R) and the setup time severity factor (η) (Lee and Pinedo 1997). Here τ is:

$$\tau = 1 - \frac{\sum d_j}{nC_{max}} \quad (2)$$

where C_{max} is the optimal makespan for the problem (if the sequence dependent setups times were disregarded, the makespan would be the sum of the p_js). The range R is obtained by the comparison of the maximal and minimal due dates, d_{max} and d_{min}, as follows:

$$R = 1 - \frac{d_{max} - d_{min}}{C_{max}} \quad (3)$$

Finally, the setup time severity factor η, is computed according to the average setup and processing times, $\bar{s}$ and $\bar{p}$ respectively:

$$\eta = \frac{\bar{s}}{\bar{p}} \quad (4)$$

While these statistics are easy to calculate, the makespan is rather involved to compute since it depends on the sequence-dependent setup times. A useful approximation to the makespan, was developed in (Pinedo 2012):

$$\hat{C}_{max} = \sum_{j=1}^{n} p_j + n\bar{s} \quad (5)$$

Approximation (5) tends to overestimate the optimal value of C_{max}, since it includes $\bar{s}$, which contemplates all the values s, while in the case in which the minimal C_{max} is sought, the optimal schedule must take into account only the smaller s values. Other approximations can be found in (Park, Kim, and Lee 2000).

As said, the value of $I_j(t,l)$ of the ATCS rule (1) depends on how K_1 and K_2 are defined by the user. In some studies, like (Lee and Pinedo 1997) and (Park, Kim, and Lee 2000) the following specifications are suggested:

$$K_1 = 1.2ln(n) - R$$

$$K_2 = \frac{\tau}{A_2\sqrt{n}}$$

being A_2 the empirical constant $A_2 = 1.8$ (further contributions just rounded it up to $A_2 = 2$ (Bektur and Saraç 2019)). On the other hand, direct empirical values have been proposed (Bektur and Saraç 2019)

$$K_1 = 4.5 + R, \quad \text{if } R > 0.5$$

$$K_1 = 6 - 2R, \quad \text{if } R \leq 0.5$$

Independently of the method of estimation of K_1 and K_2 to be used to compute the schedule, the rule depends on its definition and on how it has been treated in practice (Lee and Pinedo 1997; Park, Kim, and Lee 2000), or more recently (Xi and Jang 2012; Bektur and Saraç 2019). It would be interesting to find a method to absorb the empirical knowledge gained from the definition of these parameters in the daily production of the firm. Thus, the ATCS rule is a good example of how the structure proposed in this article, based on using a SOEK, allows to improve the values of K_1 and K_2 in time.

To incorporate K_1 and K_2 in a SOEK structure, start by defining them as $I_j(t,l)$ and z (representing the value of the objective function) and treat them as variables of the SOEK. It is worth noting that the SOEK is not applied directly to yield the scheduling problem, but to improve the process leading to its determination. The conditions imposed by the system are summarized in the statistics τ, R, η and $\hat{C}_{max}$. The objective is the minimization of $\sum w_j T_j$ ($z = \sum w_j T_j$) and the rule is ATCS. All these elements allow to implement the SOEK according to definition (Sanin et al. 2012), and to record each FDE.

According to the architecture in Figure 5, a FDE will be addressed as follows: the Integrator will compile and organize the information related to the FDE, i.e. n, τ, R, η and $\hat{C}_{max}$, as well as K_1, K_2, defined as $I_j(t,l)$ and z. This information can be organized in Comma Separated Values (csv) files, as proposed in (Sanin et al. 2019). Afterwards this information is sent to the Prognoser, which will analyze it and jointly with the Convertor will translate it to the DDNA format. When a new FDE arises, the scheduler can request the MSS to look for previous similar problems based on the values of n, τ, R, η and $\hat{C}_{max}$. The MSS with support in the DDNA can yield, accordingly, a solution.

A final comment on this case, given the speed of computation of the ATCS rule (defined by simple algebraic equations), is that this method can be applied both to release and reactive scheduling. Thus, it is a method that can be applied to rescheduling or dynamic scheduling problems to obtain schedules in real time.

5. Conclusion

Industry 4.0 prescribes the intensive digitization of production processes, leading to a substantial improvement in the availability of data and information. These new features allow the development of increasingly intelligent systems to support decision-making in production planning. In this work it is proposed to use such systems to improve the ability to solve scheduling problems by incorporating Artificial Intelligence-based complements to Manufacturing Scheduling Systems.

Scheduling problems are currently solved by human schedulers who use tools that allow them to calculate different alternative production plans. These solutions cannot be implemented directly to production systems without subsequent adjustments and calibrations. Then, it is proposed an architecture that works in parallel with the human scheduler, recording the adjustments and calibrations made. In the next iterations the system is able to provide solutions better tuned to the real problems to be solved. This is illustrated by the case of a single machine scheduling problem. The proposed structure of records allows registering the variables, restrictions, rules and functions faced at each decision event. Ihe proposed structure allows managing efficiently the database of past case studies, speeding up the resolution capacity in future situations.

As future work, it is intended to analyze more complex scheduling systems (flow shop systems, for instance), as well as the link between the knowledge repository and more advanced decision-making modules (for example, for lot-sizing problems).

References

Alavi, Maryam, and Dorothy E Leidner. 2001. "Knowledge management and knowledge management systems: Conceptual foundations and research issues." *MIS quarterly* 107–136.

Bektur, Gulcin, and Tugba Saraç. 2019. "A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server." *Computers & Operations Research* 103: 46–63.

Bożek, A, and M Wysocki. 2015. "Flexible job shop with continuous material flow." *International Journal of Production Research* 53 (4): 1273–1290.

Churchman, C West. 1971. "The design of inquiring systems basic concepts of systems and organization." .

Dolgui, Alexandre, Dmitry Ivanov, Suresh P Sethi, and Boris Sokolov. 2019. "Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications." *International Journal of Production Research* 57 (2): 411–432.

Du, Jianzhong, and Joseph Y-T Leung. 1990. "Minimizing total tardiness on one machine is NP-hard." *Mathematics of operations research* 15 (3): 483–495.

Fahey, Liam, and Laurence Prusak. 1998. "The eleven deadliest sins of knowledge management." *California management review* 40 (3): 265–276.

Ferraro, Augusto, Daniel Rossit, Adrián Toncovich, and Mariano Frutos. 2019. "Lot Streaming Flow Shop with a Heterogeneous Machine." *Engineering Management Journal* 31 (2): 113–126.

Framinan, Jose M, Rainer Leisten, and Rubén Ruiz García. 2014. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools.* Springer Science & Business Media.

Framinan, Jose M, and Rubén Ruiz. 2010. "Architecture of manufacturing scheduling systems: Literature review and an integrated proposal." *European Journal of Operational Research* 205 (2): 237–246.

Guo, Zhengang, Yingfeng Zhang, Xibin Zhao, and Xiaoyu Song. 2020. "CPS-based self-adaptive collaborative control for smart production-logistics systems." *IEEE Transactions on Cybernetics* .

Hermann, Mario, Tobias Pentek, and Boris Otto. 2016. "Design principles for industrie 4.0 scenarios." In *2016 49th Hawaii international conference on system sciences (HICSS)*, 3928–3937. IEEE.

Ivanov, Dmitry, Alexandre Dolgui, Boris Sokolov, Frank Werner, and Marina Ivanova. 2016.

"A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0." *International Journal of Production Research* 54 (2): 386–402.

Ivanov, Dmitry, Suresh Sethi, Alexandre Dolgui, and Boris Sokolov. 2018. "A survey on control theory applications to operational systems, supply chain management, and Industry 4.0." *Annual Reviews in Control* 46: 134–147.

Jessup, Leonard M, and Joseph S Valacich. 2003. *Information systems Today.* Prentice Hall.

Jung, Jason J. 2009. "Semantic business process integration based on ontology alignment." *Expert Systems with Applications* 36 (8): 11013–11020.

Kletti, Jürgen. 2007. *Manufacturing Execution System-MES.* Springer.

Kusiak, Andrew. 2017. "Smart manufacturing must embrace big data." *Nature* 544 (7648): 23–25.

Lawler, Eugene L, Jan Karel Lenstra, and AHG Rinnooy Kan. 1982. "Recent developments in deterministic sequencing and scheduling: A survey." In *Deterministic and stochastic scheduling*, 35–73. Springer.

Lee, Jay, Behrad Bagheri, and Hung-An Kao. 2015. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems." *Manufacturing letters* 3: 18–23.

Lee, Young Hoon, and Michael Pinedo. 1997. "Scheduling jobs on parallel machines with sequence-dependent setup times." *European Journal of Operational Research* 100 (3): 464–474.

Leng, Jiewu, Douxi Yan, Qiang Liu, Hao Zhang, Gege Zhao, Lijun Wei, Ding Zhang, Ailin Yu, and Xin Chen. 2019. "Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system." *International Journal of Computer Integrated Manufacturing* 1–18.

Lenstra, Jan Karel, AHG Rinnooy Kan, and Peter Brucker. 1977. "Complexity of machine scheduling problems." In *Annals of discrete mathematics*, Vol. 1, 343–362. Elsevier.

Leusin, Matheus E, Enzo M Frazzon, Mauricio Uriona Maldonado, Mirko Kück, and Michael Freitag. 2018. "Solving the job-shop scheduling problem in the industry 4.0 era." *Technologies* 6 (4): 107.

Liu, Chang, Hanying Li, Yuwei Tang, Dongtao Lin, and Jiawei Liu. 2019. "Next generation integrated smart manufacturing based on big data analytics, reinforced learning, and optimal routes planning methods." *International Journal of Computer Integrated Manufacturing* 32 (9): 820–831.

Lv, Youlong, Jie Zhang, and Wei Qin. 2019. "Simulation-based production analysis of mixed-model assembly lines with uncertain processing times." *Journal of Simulation* 13 (1): 44–54.

McKay, Kenneth N, and John A Buzacott. 2000. "The application of computerized production control systems in job shop environments." *Computers in Industry* 42 (2-3): 79–97.

McKay, Kenneth N, and Vincent CS Wiers. 1999. "Unifying the theory and practice of production scheduling." *Journal of Manufacturing Systems* 18 (4): 241.

Monostori, László. 2014. "Cyber-physical production systems: Roots, expectations and R&D challenges." *Procedia Cirp* 17: 9–13.

Ouelhadj, Djamila, and Sanja Petrovic. 2009. "A survey of dynamic scheduling in manufacturing systems." *Journal of scheduling* 12 (4): 417.

Park, Kyu Tae, Young Wook Nam, Hyeon Seung Lee, Sung Ju Im, Sang Do Noh, Ji Yeon Son, and Hyun Kim. 2019. "Design and implementation of a digital twin application for a connected micro smart factory." *International Journal of Computer Integrated Manufacturing* 32 (6): 596–614.

Park, Youngshin, Sooyoung Kim, and Young-Hoon Lee. 2000. "Scheduling jobs on parallel machines applying neural network and heuristic rules." *Computers & Industrial Engineering* 38 (1): 189–202.

Pfund, Michele, John W Fowler, Amit Gadkari, and Yan Chen. 2008. "Scheduling jobs on parallel machines with setup times and ready times." *Computers & Industrial Engineering* 54 (4): 764–782.

Pinedo, Michael. 2012. *Scheduling.* Vol. 29. Springer.

Qian, Cheng, Yingfeng Zhang, Chen Jiang, Shenle Pan, and Yiming Rong. 2020. "A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing." *Robotics and Computer-Integrated Manufacturing* 61: 101841.

Qin, W, J Zhang, and D Song. 2018. "An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time." *Journal of Intelligent Manufacturing* 29 (4): 891–904.

Qin, Wei, Zilong Zhuang, Yang Liu, and Ou Tang. 2019. "A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly." *Computers & Industrial Engineering* 138: 106115.

Rossit, Daniel Alejandro, and Fernando Tohmé. 2018. "Scheduling research contributions to Smart manufacturing." *Manufacturing Letters* 15: 111–114.

Rossit, Daniel Alejandro, Fernando Tohmé, and Mariano Frutos. 2019a. "A data-driven scheduling approach to smart manufacturing." *Journal of Industrial Information Integration* 15: 69–79.

Rossit, Daniel Alejandro, Fernando Tohmé, and Mariano Frutos. 2019b. "An Industry 4.0 approach to assembly line resequencing." *The International Journal of Advanced Manufacturing Technology* 105 (9): 3619–3630.

Rossit, Daniel Alejandro, Fernando Tohmé, and Mariano Frutos. 2019c. "Industry 4.0: Smart Scheduling." *International Journal of Production Research* 57 (12): 3802–3813.

Rossit, Daniel Alejandro, Fernando Tohmé, and Mariano Frutos. 2019d. "Production planning and scheduling in Cyber-Physical Production Systems: a review." *International journal of computer integrated manufacturing* 32 (4-5): 385–395.

Rowley, Jennifer. 2007. "The wisdom hierarchy: representations of the DIKW hierarchy." *Journal of information science* 33 (2): 163–180.

Sanin, Cesar, Zhang Haoxi, Imran Shafiq, Md Maqbool Waris, Caterine Silva de Oliveira, and Edward Szczerbicki. 2019. "Experience based knowledge representation for Internet of Things and Cyber Physical Systems with case studies." *Future Generation Computer Systems* 92: 604–616.

Sanin, Cesar, and Edward Szczerbicki. 2007. "Towards the construction of decisional DNA: A set of experience knowledge structure java class within an ontology system." *Cybernetics and Systems: An International Journal* 38 (8): 859–878.

Sanin, Cesar, Carlos Toro, Zhang Haoxi, Eider Sanchez, Edward Szczerbicki, Eduardo Carrasco, Wang Peng, and Leonardo Mancilla-Amaya. 2012. "Decisional DNA: A multi-technology shareable knowledge structure for decisional experience." *Neurocomputing* 88: 42–53.

Shafiq, Syed Imran, Cesar Sanín, and Edward Szczerbicki. 2014. "Set of experience knowledge structure (SOEKS) and decisional DNA (DDNA): past, present and future." *Cybernetics and Systems* 45 (2): 200–215.

Tao, Fei, Qinglin Qi, Ang Liu, and Andrew Kusiak. 2018. "Data-driven smart manufacturing." *Journal of Manufacturing Systems* 48: 157–169.

Tasgetiren, M Fatih, Quan-Ke Pan, and Yun-Chia Liang. 2009. "A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence dependent setup times." *Computers & Operations Research* 36 (6): 1900–1915.

Vepsalainen, Ari PJ, and Thomas E Morton. 1987. "Priority rules for job shops with weighted tardiness costs." *Management science* 33 (8): 1035–1047.

Vernon, C. 2001. "Lingering amongst the lingerie: An observation-based study into support for scheduling at a garment manufacturer." In *Human Performance in Planning and Scheduling*, edited by B. Mac Carthy and J. Wilson, 135–163. London: Taylor & Francis.

Vieira, Guilherme E, Jeffrey W Herrmann, and Edward Lin. 2003. "Rescheduling manufacturing systems: a framework of strategies, policies, and methods." *Journal of scheduling* 6 (1): 39–62.

Wang, Junliang, Peng Zheng, Wei Qin, Tengda Li, and Jie Zhang. 2019. "A novel resilient scheduling paradigm integrating operation and design for manufacturing systems with uncertainties." *Enterprise Information Systems* 13 (4): 430–447.

Wang, Peng, Cesar Sanín, and Edward Szczerbicki. 2013. "Prediction based on integration of decisional DNA and a feature selection algorithm RELIEF-F." *Cybernetics and Systems* 44 (2-3): 173–183.

Wang, Wenbo, Yingfeng Zhang, and Ray Y Zhong. 2020. "A proactive material handling method for CPS enabled shop-floor." *Robotics and Computer-Integrated Manufacturing* 61: 101849.

Xi, Yue, and Jaejin Jang. 2012. "Scheduling jobs on identical parallel machines with unequal future ready time and sequence dependent setup: An experimental study." *International Journal of Production Economics* 137 (1): 1–10.

Yu, Ying, Xin Wang, Ray Y Zhong, and GQ Huang. 2017. "E-commerce logistics in supply chain management." *Industrial Management & Data Systems* 117 (10): 2263–2286.

Zhang, Kai, Ting Qu, Dajian Zhou, Hongfei Jiang, Yuanxin Lin, Peize Li, Hongfei Guo, Yang Liu, Congdong Li, and George Q Huang. 2020. "Digital twin-based opti-state control method for a synchronized production operation system." *Robotics and Computer-Integrated Manufacturing* 63: 101892.

Zhang, Yingfeng, Zhengang Guo, Jingxiang Lv, and Ying Liu. 2018. "A framework for smart production-logistics systems based on CPS and industrial IoT." *IEEE Transactions on Industrial Informatics* 14 (9): 4019–4032.

Zhong, Ray Y, Xun Xu, Eberhard Klotz, and Stephen T Newman. 2017. "Intelligent manufacturing in the context of industry 4.0: a review." *Engineering* 3 (5): 616–630.