International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: http://www.tandfonline.com/loi/tprs20

Industry 4.0: Smart Scheduling

Daniel Alejandro Rossit, Fernando Tohmé & Mariano Frutos

To cite this article: Daniel Alejandro Rossit, Fernando Tohmé & Mariano Frutos (2018): Industry 4.0: Smart Scheduling, International Journal of Production Research

To link to this article: https://doi.org/10.1080/00207543.2018.1504248

Published online: 03 Aug 2018.

Submit your article to this journal

View Crossmark data

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=tprs20

International Journal of Production Research, 2018
https://doi.org/10.1080/00207543.2018.1504248

Taylor & Francis
Taylor & Francis Group
Check for updates

Industry 4.0: Smart Scheduling

Daniel Alejandro Rossit [ab*], Fernando Tohmé [bc] and Mariano Frutos[ad]

[a]*Department of Engineering, Universidad Nacional del Sur, Bahía Blanca, Argentina;* [b]*INMABB UNS CONICET, Departamento de Matemática, Bahía Blanca, Argentina;* [c]*Department of Economics, Universidad Nacional del Sur, Bahía Blanca, Argentina;* [d]*IIESS UNS CONICET, Departamento de Economia, Bahía Blanca, Argentina*

(*Received 26 January 2018; accepted 28 June 2018*)

Smart Manufacturing and Industry 4.0 production environments integrate the physical and decisional aspects of manufacturing processes into autonomous and decentralised systems. One of the main aspects in these systems is production planning, in particular scheduling operations on machines. We introduce here a new decision-making schema, *Smart Scheduling*, intended to yield flexible and efficient production schedules on the fly, taking advantage of the features of these new environments. The ability to face unforeseen and disruptive events is one of the main improvements in our proposed schema, which uses an efficient screening procedure (Tolerance Scheduling) to lessen the need of rescheduling in the face of those events.

Keywords: scheduling; Industry 4.0; Smart Manufacturing; Cyber-Physical Systems; decision-making; dynamic scheduling; inverse scheduling

Introduction

The last few years have witnessed the emergence of a fourth industrial revolution, referred to as *Industry 4.0* (Wang, Ong, and Nee 2017; Xu 2017). It amounts to a paradigm change in manufacturing processes, based on the heavy use of automated tools (Tao et al. 2017). Hermann, Pentek, and Otto (2016) present all the practitioner and academic information on this issue, advancing the following definition of Industry 4.0:

> Industry 4.0 is a collective term for technologies and concepts of value chain organization. Within the modular structured Smart Factories of Industry 4.0, Cyber-Physical Systems (CPS) monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over the Internet of Things (IoT), CPS communicate and cooperate with each other and humans in real time. Via the Internet of Services (IoS), both internal and cross organizational services are offered and utilized by participants of the value chain.

As indicated by this definition, CPS – which integrate computational systems with physical processes (Lee 2008) – are essential components of this new paradigm. Wang, Törngren, and Onori (2015) present a thorough analysis of the impact of CPS on the future manufacturing environments, based on their ten more salient characteristics. Many of them involve the decision-making process in production settings, endowing CPS with the ability to respond, rather autonomously and flexibly, to unforeseen situations. CPS have the additional advantage of generating virtual replicas of the production process (Lee, Bagheri, and Kao 2015), facilitating its remote administration through Cloud Computing (Wang et al. 2014; Gao et al. 2015). This, in turn, allows viewing the production environment as the embodiment of Cloud Manufacturing (Xu 2012; Wang and Wang 2014; Wang, Gao, and Fan 2015; Liu et al. 2018). These and other applications of CPS have led to the wider concept of Cyber-Physical Production Systems (CPPS) (Monostori 2014), representing the ensemble of subsystems connected to the environment and among them in these enhanced Industry 4.0 settings. One of the benefits of CPPS is the possibility of linking directly the shop floor with a higher level Decision Support System (DSS) (Rossit and Tohmé 2018), providing real-time data to the DSS as well as giving the shop floor the ability to rapidly adapt to the output of the DSS.

These new manufacturing structures will induce changes in the way production planning is carried out. We propose here a smart approach to solving production scheduling problems, which are known for being NP-hard (Pinedo 2012), even for small instances, as for example Flow Shop cases with at least four machines (Garey, Johnson, and Sethi

*Corresponding author: Email: daniel.rossit@uns.edu.ar

© 2018 Informa UK Limited, trading as Taylor & Francis Group

1976). Since schedules are usually chosen for short time frames, the planning process has to be repeated frequently (sometimes even several times in a single week). The quality of the solutions has a direct economic impact on the benefits of companies, and thus on their long-term ability to thrive in competitive markets (Framinan, Leisten, and García 2014). This intrinsic criticality of scheduling processes becomes even more salient in their incorporation into the Smart Manufacturing processes of Industry 4.0. As pointed out by Monostori (2014), scheduling processes constitute one of the main challenges in the design of CPPS.

Manufacturing Scheduling Systems (MSS) require computer support to make decisions that human schedulers carry expertly out (Pinedo 2012). The MSS and scheduler ensemble is capable of solving all scheduling problems, either standard or dynamical. An advantage of this arrangement is that it combines computation speed with the expertise of the human scheduler, who can assess the relevance of unexpected events and discard those that can be actually discarded (Ouelhadj and Petrovic 2009). This means that full automatisation of the scheduling process requires endowing the CPPS with tools to replace the MSS-scheduler combination. The hardest part in this replacement is to incorporate the human decision-making process, which constitutes a black box for the MSS. Our proposal is to adopt the novel approach of Inverse Scheduling processes (Brucker and Shakhlevich 2011). The latter is based on the idea of adapting the environment or scenario parameters in order to ensure that a schedule is optimal, given the assumed technological constraints of the process. We use a similar concept, addressing the *Tolerance Scheduling problem*, which amounts to identifying the range of scenarios in which a given schedule keeps being optimal or at least acceptable in practice.

The new configuration of scheduling procedures presented in this paper, which we call *Smart Scheduling*, combines the concepts and tools from Smart Manufacturing with more traditional ideas on scheduling. Smart Scheduling takes advantage of the flexibility of CPPS and the ability of DSS to provide relatively fast solutions to dynamic scheduling problems. A problem in these decentralised settings is that the real-time communication between the shop floor and its corresponding DSS either increases the noise in the system or overwhelms it. Smart Scheduling provides tools to overcome this problem and to reprogram schedules applying Inverse Optimization and Inverse Scheduling techniques (Ahuja and Orlin 2001; Brucker and Shakhlevich 2011).

In section 2, we present the main concepts in the theory of scheduling, providing the backbone for our approach. Section 3 analyses the features of Industry 4.0 processes that are relevant to decision-making in this setting. Then we present our ideas on how production planning should be integrated into Industry 4.0 processes. Finally, we present Smart Scheduling and discuss its main component, a new kind of scheduling problem posed by this schema: the *tolerance scheduling problem*.

Scheduling problems

The choice of a schedule of operations is part of a production planning process. More precisely, it is the last stage of planning before the actual execution of the plan (Pinedo 2012). Scheduling involves the allocation of the available production resources in a workflow generated in a previous planning stage. The choice of a schedule demands a detailed description of the production process and amounts to handle a large volume of information (Framinan, Leisten, and García 2014; Rossit, Tohmé, and Frutos 2018). As it is intuitively evident, these decision problems have a strong combinatorial nature and consequently a high complexity.

Formally, a scheduling problem is the allocation of a family N of jobs, $N = \{1, 2, \dots, n\}$ on a set M of machines, $M = \{1, 2, \dots, m\}$. 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 $p_{ij} \in \mathbb{N}$ on machine i. Each job j will have associated 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.

Dynamic scheduling

Dynamic scheduling has been widely investigated in the past (Vieira, Herrmann, and Lin 2003; Ouelhadj and Petrovic 2009), but the increasing possibility of getting real-time information has renewed the interest on this problem. The difference with the general problem described above is that it involves planning while unforeseen events disrupt the execution of a schedule. So, for instance, if a machine goes out of service because of some failure, the operations assigned to it have to be redirected to other machines, rescheduling the production process. This problem is far from trivial and, if it inadequately addressed, may cause large losses (Vieira, Herrmann, and Lin 2003).

The disrupting events can have two main origins, either problems with the resources in the shop or problems with the jobs to be carried out. Among the former, we have machine breakdowns, operator illnesses, unavailability or tool failures, loading limits, delays in the arrival or shortage of materials and defective materials, among others. The events involving the jobs can be rush jobs, job cancellations, due date changes, early or late arrival of jobs, changes in job priorities, changes in job processing times, etc.

There exist different strategies to address the rescheduling task in the face of events disrupting the production process. One class of strategies is purely reactive, making decisions all along the apparition of the events, for instance by dispatching jobs when production orders reach the shop floor, not starting with an initial schedule. Another kind of strategy has a predictive-reactive nature and is more commonly used in fabrication systems. Predictive-reactive scheduling is a multi-stage process in which schedules are revised in response to real-time events. In the first (predictive) stage, a schedule is generated as a solution to the problem without considering possible interruptions. In the next stages (reactive), the original schedule is modified to address unforeseen events, assuming that no further events will appear. This is repeated every time a rescheduling is required (Li, Pan, and Mao 2015).

There exist strategies that start from assuming uncertainties in the decision-making process. This is the case of Robust Predictive-Reactive Scheduling (Al-Hinai and ElMekkawy 2011). This strategy makes the original schedule more robust as to ensure a lower impact on the performance of eventual disruptions and the ensuing reschedules. Alternatively, Robust Proactive Scheduling provides each operation with an extra processing time, shielding the schedule from a certain type of uncertainty, reducing the number of potential reschedules.

An important aspect of Dynamic Scheduling is that it takes into account the fact that the production process will be already being carried out when the disruptions happen. Reschedules that disregard this aspect may incur in heavy modifications that may generate unduly losses (arising, for example, from stopping ongoing jobs, modifying allocations already being implemented or moving jobs from a machine to another, etc.). A way to reduce such unwarranted changes is by incorporating into the objective function the minimisation of the number of jobs that have to change the starting dates originally scheduled (Katragjini, Vallada, and Ruiz 2013).

Inverse and reverse scheduling

The analysis of the Inverse Scheduling problem uses tools developed to address Inverse Optimization problems (Ahuja and Orlin 2001). The general inverse optimisation problem consists in inferring the values of the parameters of a model (cost coefficients, constraint values, etc.) up from observable values (for instance, the optimal values of the optimal decision variables). This means that, given a solution, the parameters of the model are determined, for which the solution is optimal. The variables in inverse optimisation problems (adjustable parameters) can be either the coefficients of the objective function (e.g. the costs in the allocation problem) or coefficients of the constraint inequalities. Inverse problems in scheduling tend to be in the latter class, where the adjustable parameters are either processing times (Koulamas 2005) or due dates or release times (Brucker and Shakhlevich 2009).

While in the traditional scheduling problem all the parameters are known, in the inverse scheduling problems those parameters are assumed to be unknown and have to be determined in order to make optimal a given schedule (Brucker and Shakhlevich 2011). The determination of the values of the unknown parameters is usually restricted to certain intervals. Brucker and Shakhlevich (2009), for instance, analyse schedules on a single machine with the goal of minimising the maximum tardiness of jobs. The tardiness of a job j is the extra time required to finish it with respect to its delivery date. Given a delivery date d_j and a schedule π, tardiness is given by:

$$L_j(\pi, d) = C_j(\pi) - d_j$$

Here C_j is the time needed to finish job j under schedule π, while d is the vector of delivery dates of all the jobs. Then, maximum tardiness is obtained according to:

$$L_{\max}(\pi, d) = \max_{j \in N} \{L_j(\pi, d)\}$$

The classic scheduling problem involves finding π^* such that $L_{\max}(\pi, d)$ is minimal, i.e.:

$L_{\max}(\pi^*, d) \leq L_{\max}(\pi, d)$, for any schedule π.

In the inverse scheduling setting, the processing time p_j (in a single machine case we do not need the extra sub-index j), and the delivery date d_j obtain up from schedule π. More precisely, assume that in our single machine example only delivery dates can be adjusted, say that for each job j the delivery date can be in an interval, $d_j \in [\underline{d}_j; \bar{d}_j]$. Thus, the adjusted delivery date $\hat{d}_j$ must be such that $\hat{d}_j \in [\underline{d}_j; \bar{d}_j]$ for each job j, yielding a vector $\hat{d} = (\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n)$. The Euclidean discrepancy $\|\hat{d} - d\|$ with respect to the original d has to be minimal and π optimal. Thus, the inverse scheduling problem is here:

$$\min \|\hat{d} - d\|$$

$$\text{s.t.} \qquad L_{\max}(\pi, \hat{d}) \leq L_{\max}(\sigma, \hat{d}),$$

For any schedule σ, $\underline{d}_j \leq \hat{d}_j \leq \bar{d}_j, j \in N$.

Other metrics can be applied to measure the deviations to be minimised, possibly modifying the results (Ahuja and Orlin 2001).

Another problem arises when the goal is to achieve a fixed value of the objective function. Starting from an initial solution, the initial values of the parameter are adjusted as to reach a given value of the objective function. This problem is known as Reverse Scheduling problem, which in the case of minimising maximum tardiness on a single machine is as follows. We seek to find $\hat{d}_j \in [\underline{d}_j; \bar{d}_j]$, such that the value L^* of the objective function $L_{\max}$ remains the same or gets improved. That is:

$$\min \|\hat{d} - d\|$$

$$\text{s.t.} \qquad L_{\max}(\sigma, \hat{d}) \leq L^*,$$

For any schedule σ, $\underline{d}_j \leq \hat{d}_j \leq \bar{d}_j, j \in N$.

Other parameters are subject to choice as to solve inverse and reverse scheduling problems, a sample of which can be found in Heuberger (2004).

Manufacturing scheduling systems

Given the combinatorial nature and the complexity of most scheduling problems, DSS are usually needed to support the process of decision-making (Dolgui et al. 2018). These systems are the MSS and constitute a variant of Business Information tools, i.e. information systems supporting business functions, such as those provided by the SIEMENS PREACTOR business unit or SAP's integrated management systems (Bożek and Wysocki 2015). Framinan and Ruiz (2010) present a guideline for the design, implementation and testing of an MSS. The model of Pinedo (2012) provided a general description of an MSS (Figure 1).

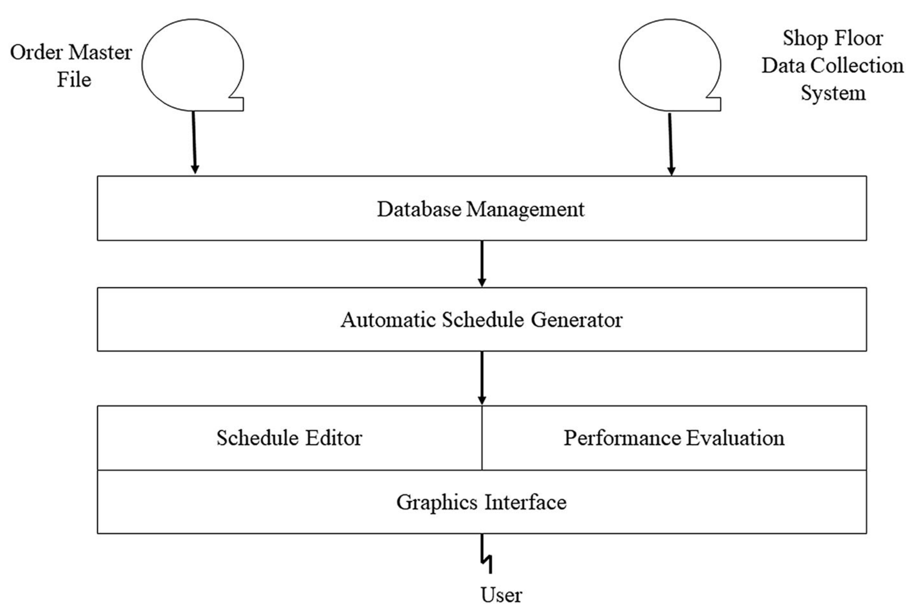

Figure 1. Scheduling System (Pinedo 2012).

The system is constituted by the following components: a Database Management module, an Automatic Schedule Generator, a Schedule Editor and a Performance Evaluator. Each of these last two components has its own Graphic User Interface (GUI). The Database Management module manages the information required to develop a production schedule. This information is generated on the basis of the production orders and the master production programs, as well as from shop floor data, which allows monitoring the state of the physical aspects of production. The output of the Database Management module feeds into the Automatic Schedule Generator.

The MSS, represented in Figure 1, is intended as a decision-making aide to the scheduler or the final user. The goal is to produce a working schedule and also address events that arise in the dynamics of the production process (Pinedo 2012). The main tasks faced by the users of the system are the allocation of jobs to resources (in general, machines), handle problems affecting schedules (like changes in resources, dates, quantities, etc.) and anticipate future problems with the schedule (Framinan, Leisten, and García 2014). The field study of McKay and Buzacott (2000) showed that human schedulers usually follow a 'script', independently of the production field in which they operate. It starts by evaluating the current situation, looking for critical issues or sources of conflict, as for instance a job that takes longer than planned or wrong uses of resources. Once identified, the scheduler has to determine whether to run a rescheduling process or reassign resources. Once done that, it updates the information on the schedule and runs again an analysis of possible critical issues.

Industry 4.0 and Smart Manufacturing

In this section, we will briefly discuss production planning, and in particular, scheduling, in Industry 4.0 and Smart Manufacturing environments. We will first review the emerging technologies in this new paradigm, focusing on the new production resources embodied in CPS and CPPS.

Industry 4.0: technologies

The main difference between Industry 4.0 and its predecessors is that, instead of the traditional hierarchical and centralised structures, it exhibits schemes in which autonomous agents interact in decentralised architectures. These agents are connected among them and with decision centres. The technologies that are mostly relevant for the process of decision-making are Cloud Computing, Internet of Things (IoT), Big Data and RFID connections.

Cloud computing allows virtualising and scaling resources in a dynamic way. Its use provides firms with the possibility of getting resources as they are needed, without incurring in sunken costs and paying only for the resources actually used (Ahn, Park, and Hur 2018; Wang and Wang 2018a). IoT is the portmanteau expression for the digital connection of objects to the Internet. It involves the different technologies that allow the smart integration of objects online and thus to follow remotely the state of execution of work orders, collecting data and information in real time (Wang and Wang 2018b). Big data, in turn, refers to the techniques for processing large and inhomogeneous databases collected online. Novel computational procedures permit the detection of patterns where no traditional methods could yield useful insights or even be applied. In the case of manufacturing, Big Data methods allow accessing and processing large amounts of data generated in production processes (Wang and Wang 2018b). RFID refers to procedures to store and recover data remotely using RFID labels, cards or transponders. The identity of an object (akin to its idiosyncratic serial number) can be transmitted to others through radio waves. This technology provides a way to exchange relevant information between fast moving objects at long distances (Wang and Wang 2018b).

CPS and CPPS

As briefly discussed in the Introduction, CPS are some of the main components of Industry 4.0 systems. CPS facilitate the confluence of physical and virtual spaces, integrating computational and communication processes in interaction with physical processes, adding new capabilities to physical systems (Wang, Törngren, and Onori 2015). Unlike traditional embedded systems, which tend to be independent, CPS feature a network of interactive I/O physical elements, similar to networks of sensors. Later years have witnessed great advances in this area. New intelligent CPS will spur innovation and competition in different industries (aerospace, automobile, chemical, energy, infrastructure, transportation, etc.). A relevant instance of CPS is constituted by intelligent manufacturing lines, in which a single machine can carry out a variety of procedures communicating with the other components (Wang and Wang 2018b).

CPS with manufacturing-specific implementations have given rise to CPPS. According to Monostori (2014), CPPS consist of autonomous and cooperating elements and subsystems interconnected in such way that, depending on the setting, cover all the stages of the production process, from the shop floor to the logistic networks. One of the main challenges posed by these systems is the need to develop robust approaches to scheduling, in order to face adequately to the different and unforeseen stresses on distributed production processes.

Industry 4.0: decision-making processes

To analyse how decision-making processes can be distributed in Industry 4.0 environments, applying the previously discussed technologies, we use as a reference the ANSI/ISA 95 standard. This standard provides a widely accepted representation of the architecture of a firm and its different levels of decision-making.

ANSI/ISA 95

The ANSI/ISA 95 is a standard that can provide a framework for an automated interface between production facilities and control systems. It can yield a common ground for the communication of all the participants in a production process, and gives a representation of how information can be modelled and used. It organises the different levels of decision-making hierarchically. It is based on the 'Purdue Enterprise Reference Architecture' (PERA) which distinguishes five levels, as shown in Figure 2. Level 0 is associated with the physical process of manufacturing. Level 1 involves the intelligent devices that measure and manipulate the physical process are located. Typical instruments at this level are sensors, analysers, effectors and related instrumentation. Level 2 represents the control and supervision of the underlying activities. Systems acting on ISA-95 Level 2 are Supervisory Control and Data Acquisition (SCADA), Programmable Logic Controllers (PLC), Distributed Control Systems (DCS) and Batch Automation Systems. Then, level 3 involves the management of the operations and the production workflow to produce the desired products. Some of the systems comprised at this level are Batch management; manufacturing execution/operations management systems (MES/MOMS); laboratory, maintenance and plant performance management systems; data historians and related middleware. This level has special importance for our work since it is where the scheduling process takes place. Finally, level 4 is associated with the business activities of the entire firm. This architecture represents, in a synthetic way, the different activities and functions of a production system. Besides, it establishes the way in which the different levels communicate; in particular that in traditional productions settings, each level interacts only with its adjacent levels (Rossit and Tohmé 2018).

Decision-making in CPPS

As discussed in Rossit and Tohmé (2018), CPPS will have an impact on decision-making activities in the area of industrial planning and control. This will be due to the ability of CPPS to carry out a wide spectrum of activities, ranging from the physical operations of production (level 0) to planning, evaluating and managing the entire production process (level 3), by controlling the actions and systems on levels 1 and 2 (i.e. the measurement and sensing instruments as well as the

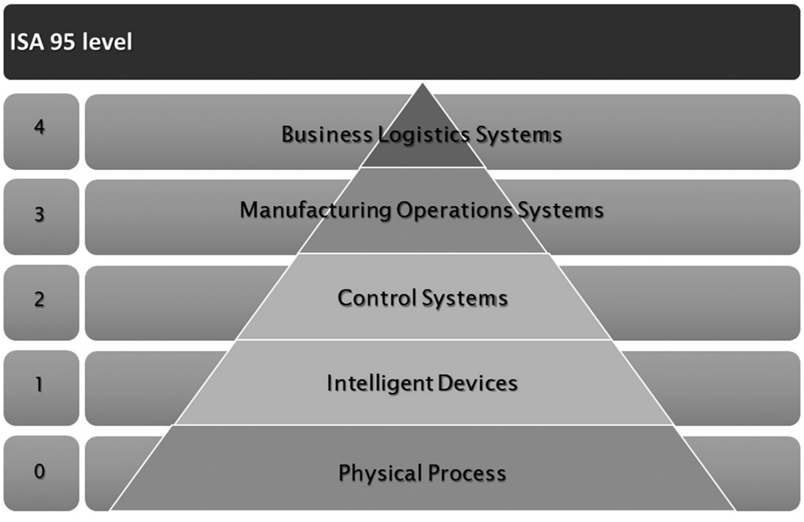

Figure 2. Control structure ANSI/ISA 95 (Rossit and Tohmé 2018).

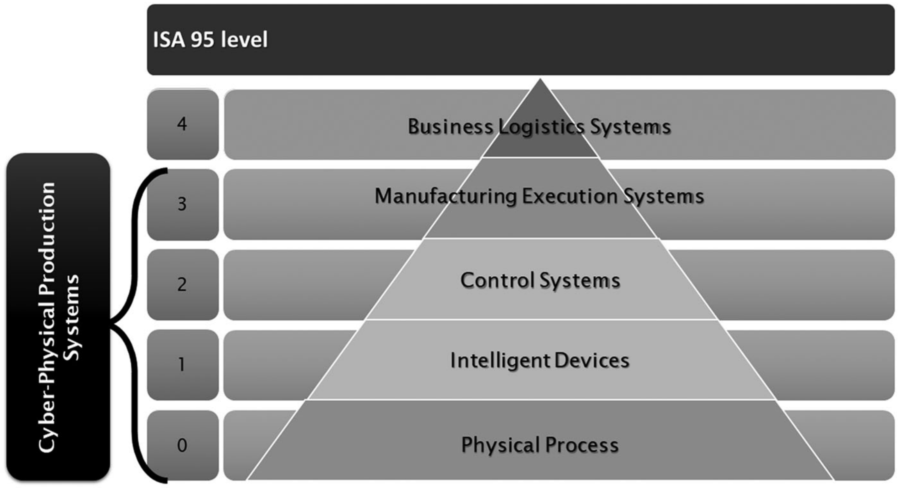

Figure 3. Levels of ANSI/ISA95 integrated into CPPS.

control systems). This approach is illustrated in Figure 3, in which the levels of ISA 95 that should be incorporated to CPPS are highlighted.

Some of the direct benefits of this integration of functionalities are, for instance, the increased flexibility to respond to unexpected events, or the faster transmission of information through the entire system. These advantages are due to the fact that CPPS can translate the data obtained at level 1 to the higher order language used at level 3, eluding the adjacency limitations inherent in PERA, generating answers faster to unforeseen events.

This, in turn, will affect directly the way in which decisions are made in production planning, which in terms of PERA will be managed by both Enterprise Resource Planning (ERP) and the CPPS. Figure 4 shows this.

Figure 4 indicates that only the decisions at the aggregate level (as for instance the goals of the company) will be handled by ERP systems, already adapted to smart manufacturing environments. All other decisions will automatically be made and executed by CPPS. In this way, current Manufacturing Execution Systems (MES) will be absorbed by CPPS, which will also take care of integrating the dispatch of work orders and their schedule in the shop floor. This will improve the quality of the information at this level, increasing the flexibility and the ability to respond to changing circumstances (Rossit and Tohmé 2018).

The ensuing structure, jointly handled by ERP and CPPS in a distributed fashion, modifies the traditional hierarchical and centralised view on production planning. Of course, this will also affect Scheduling, since decisions will now be made in a distributed and collaborative way (Wang and Wang 2018c). Despite this, the literature still treats the problem in the

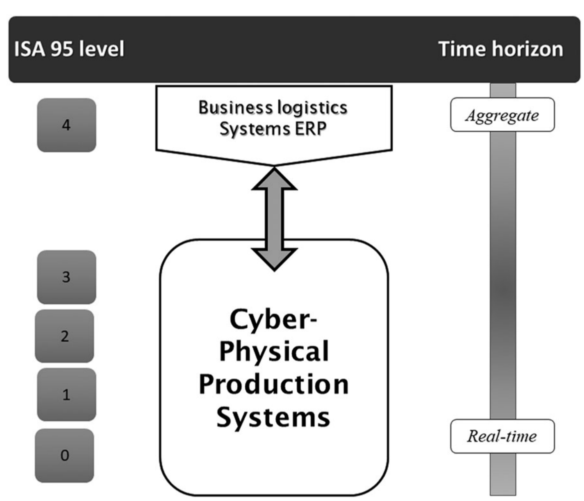

Figure 4. Distribution of ISA 95 levels between ERP and CPPS. The representation of time is drawn from the model of the Manufacturing Enterprise Solutions Association (MESA) International.

traditional and static way. The scheduling community will have to accept the challenges presented by the new paradigm, developing strategies and methods accordingly. The incorporation of more complex and flexible production structures will be a substantial contribution, as in non-permutation programming of fabrication cells (Rossit et al. 2016, Rossit, Tohmé, and Frutos 2018).

A good deal of the decisions made by ERP systems (as inventory control, management of databases, handling information about suppliers, etc.) will be managed by CPPS. But we leave them separated as to indicate at what point the system becomes autonomous and up to which human interventions may be needed, particularly in the area of production planning. The linkage with human decision-makers will be at the aggregate or strategic level. They will define the goals and guidelines for the firm and the system. An ERP system will get them and will translate these guidelines for the rest of the system, in particular to the CPPS that handle the production system. The latter are thus not completely autonomous since they keep an open loop with the ERP system, at least on production planning (see section 2.4.1.1 in Framinan, Leisten, and García 2014).

Smart Scheduling

In this section, we present a novel notion, namely that of *Smart Scheduling*. First, we introduce a mechanism to mitigate the problems induced by the real-time autonomous behaviour of components of the system. This will lead to the full scheme of Smart Scheduling.

The tolerance scheduling problem

The direct connection between the decision-making processes with the shop floor makes rescheduling processes event-driven. This means that rescheduling is triggered by events modifying the setting of the initial schedule. To automatise this process, we need criteria for the identification of the events that lead to reschedules, which have to be defined beforehand and must thus be related to the essential features of the event, be it originated in the production resources or in the jobs to be carried out. Once identified, an event may belong to the class that triggers reschedules, but even so, its magnitude may not merit starting the rescheduling process. This becomes an additional source of instability, exposing the production process to many uncertainties affecting the plan (Pinedo 2012; Framinan, Leisten, and García 2014). To react at each event disturbs the shop floor and affects the efficiency of the system. Our take is to address a new scheduling problem in which the solution is tolerant to a range of events that by themselves are deemed as triggers of rescheduling.

This *Tolerance Scheduling* problem starts with an initial solution (optimal or near-optimal). The goal is to generate a range of tolerances, mainly for the parameters of the model. As with the specification of tolerances for manufactured goods that allow for a range within which the good is still considered appropriate, here we allow for a certain degree of imperfection in the plan actually carried out. Consider for instance situations in which the actual processing times differ from the specifications used to solve the original scheduling problem. This event has an impact on the performance of the production process (e.g. worsening the makespan), which would call for rescheduling the plan. But it is worth to ponder whether the gains of doing this outweigh the costs of rescheduling.

Consider again the single-machine scheduling problem of Brucker and Shakhlevich (2009). In the Inverse Scheduling problem, we seek to find the adjusted delivery dates of each job, d_j in order that a given schedule π becomes optimal. In the Reverse Scheduling, instead, we seek to make minimal adjustments to the delivery dates and the schedule as to ensure a certain range of values for the objective function. In our case, we seek tolerances for the parameters ensuring that the original schedule remains acceptable and thus no rescheduling is necessary.

Formally, given an optimal or near-optimal schedule π, $F(\pi) \approx L^*$, and the families of parameters d_j and p_{ij}, we seek a maximal interval of variations for them, we also incorporate an *inertia factor*, δ, expressing the weight given to the stability of the system. A high δ indicates that the design favours a high stability (high inertia), meaning that few events can trigger reschedules. Then:

$$\max \|\hat{d} - d\|$$

$$\text{s.t.} \quad L_{\max}(\pi, \hat{d}) \leq L_{max}(\sigma, \hat{d}) \cdot (1 + \delta),$$

$$L_{\max}(\pi, \hat{d}) \leq L^* \cdot (1 + \delta),$$

For any schedule σ, $\underline{d}_j \leq \hat{d}_j \leq \bar{d}_j$, $\delta \geq 0, j \in N$.

That is, the goal is to maximise the distance between the d parameters while ensuring that schedule π improves the original objective function up to an inertia factor $\delta \geq 0$. This provides a tool that not only detects possible rescheduling events but also determines whether or not to proceed with the rescheduling process. The choice of δ is not arbitrary: it must be proportional to the weight given to the *inertia* of the production process. That is, if the idea is to reschedule only at high levels of disruption (high inertia), δ must be large. On the contrary, a low inertia system should be more ready to react, which requires a lower δ.

This procedure is rather easy to automatise, providing another tool to be added to the DSS embedded in the CPPS, making the latter more prone to autonomous behaviour. The value of δ should, in that case, be set at the design stage.

The scheme of Smart Scheduling

Smart Scheduling is introduced as a framework for scheduling in production planning by using the tools of Smart Manufacturing and Industry 4.0 environments. As shown in Figure 4, this is handled mainly by a CPPS. The goal of Smart Scheduling is to automatise the solution to the scheduling problems in the integrated frame of CPPS.

We have to prove that the functionalities of a classical MSS (shown in Figure 1) which is part of the MSS-scheduler ensemble in an autonomous CPPS are captured in our approach. First, let us analyse the MSS in itself, then the scheduler and finally their combination. The functionalities provided by the MSS are easily integrated into a CPPS, since they can be connected to different business functions. CPPS are by design computer (and physical) systems wired in such way as to support all those functions (Monostori 2014). This amounts to say that CPPS can replace and even improve over MSS, being able to run quality analyses or failure diagnoses, increasing the global efficiency of the system.

This is not the case of the skills of the scheduler, which are not that easy to integrate into the basic design of CPPS (Lee, Bagheri, and Kao 2015). To extend the functionalities of CPPS in this respect, we introduce the Smart Scheduling framework, which fundamentally addresses the Tolerance Scheduling Problem. Our proposal provides reliable optimisation-based tools to assess the criticality of events, in terms of their nature and magnitude. Endowed with this ability, the CPPS can trigger reschedules only when the goals of the schedule get considerably affected, improving the resilience of the system and decreasing its sensitivity to the noise generated by the environment.

Smart Scheduling follows the logic of dynamic scheduling, as is proposed in Figure 5. In a first stage, the problem, either in the standard or the stochastic version, is solved, yielding an initial schedule. Next, the tolerances are set to be used in solving the Tolerance Scheduling problem. In a second stage, the production process is started, following the original schedule, until a disruptive event is detected. The event is analysed to determine if it requires triggering a rescheduling procedure. If not, the production process continues. But if, by its very nature, the event belongs to the class of those that might trigger reschedules, the Tolerance Scheduling procedure is invoked. If the current schedule falls within the range of the tolerance, the

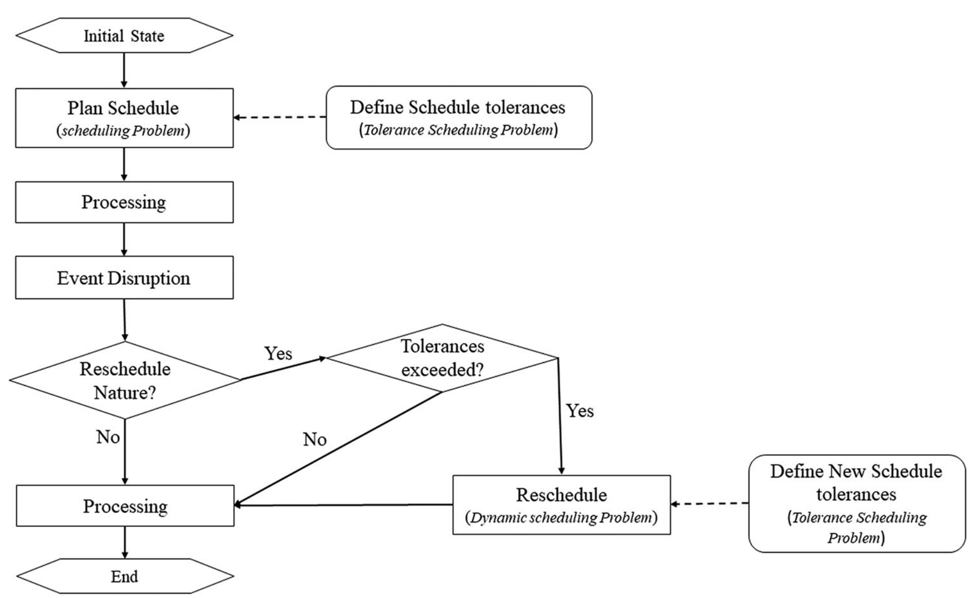

Figure 5. Smart Scheduling Schema.

production process continues. Otherwise, a new schedule is generated to address the disruption caused by the event. Then, once the schedule is obtained by applying the dynamic scheduling strategy, new tolerances are set for a future eventual solution of the Tolerance Scheduling.

Example of Smart Scheduling implementation

In order to illustrate how Smart Scheduling works, showing that it embodies the functionalities of an MSS-scheduler, we present an example. Suppose that an event affects a running schedule, namely a delay in processing an operation O_{ij}, which had an expected processing time p_{ij}, but actually became $p_{ij}^d = p_{ij} + \varepsilon$, with $\varepsilon > 0$.

The MSS-Scheduler addresses this problem as follows. The MSS sends an alert signal to the dashboard (on the GUI interface). It also informs about the event, indicating that it amounts to a deviation from the processing time of O_{ij}. In turn, the MSS indicates the magnitude ε of the deviation and offers the scheduler the possibility of reprogramming the full schedule or just a part of it. The scheduler uses the Schedule Editor and the Performance Evaluator (Figure 1) to quantify the deviation and assess the need of potential reschedules. At this stage, the expertise of the scheduler becomes crucial. She has to decide whether to ask the MSS for a new schedule or discard the event. This is, in fact, very similar to what happens when the MSS used is the SIEMENS PREACTOR.

Now let us analyse the same problem, implementing the Smart Scheduling scheme. Since it incorporates the same functionalities as an MSS, it will do the same up to sending an alarm and compute the deviation ε. Smart Scheduling then recalls the tolerances obtained solving the Tolerance Scheduling Problem. These tolerances determine whether to trigger or not a reschedule. A design factor in the computation of tolerances, parameter δ, represents the reactivity or inertia of the system in the face of deviations like ε.

Features and challenges

The Smart Scheduling scheme makes the best possible use of the features of Industry 4.0 and Smart Manufacturing environments and seems an appropriate contribution to their design. It is grounded on the dynamic scheduling paradigm, which is widely used in industries facing complex scheduling problems. Then, the adoption of Smart Scheduling will allow industries to profit from the cumulative experience in their fields (for instance, facilitating the selection of appropriate rescheduling strategies). In turn, the adoption of tolerance scheduling allows making intuitive and natural decisions, seen from the point of view of the requirements of manufacturing industries, overcoming the resistance to the implementation of Smart Manufacturing and Industry 4.0 environments, usual among personnel in traditional firms.

This said, there exist challenges that an automatised scheduling procedure like Smart Scheduling has to face up. Let us review some of them:

- *Data and information management*. Scheduling processes require the use of a large amount of data on different aspects of the manufacturing setting, as for instance information about work orders, the state of the machines, the possible processing paths, etc. Any acceptable solution to the scheduling problem can only be found if data can be efficiently collected from the autonomous and decentralised components of the system. Furthermore, this information must be available, in the right format, from across all the system, demanding a careful design of the whole data acquisition and processing operations associated with the production process.
- *Solution procedures.* As already pointed out, scheduling problems are highly complex. Thus, the availability of the best possible computational procedures is critical in autonomous and decentralised production environments. Smart Manufacturing will achieve best results if it uses Cloud Computing methods, running routines in high-performance computing facilities. This will allow speeding up the computation of solutions to the scheduling problem. Access to these computational resources must be ensured as to make the system agile and flexible. Furthermore, the distributed nature of Cloud Computing must be profited to obtain alternative solutions in parallel, instead of a single one that may not be efficient. But this open and decentralised use of computational resources will require the protection from cyber attacks and other online dangers.
- *Modularity and Scalability.* The very nature of Smart Manufacturing and Industry 4.0 leads to the use of plug and play physical and decisional components, changing easily the configuration of the system. Smart Scheduling, being an ongoing process, becomes harder due to this additional source of instability. It must be ensured that as new components are added to the system, the information about them becomes readily available, in the right format, to the decision-making procedures.

Conclusions

We have examined here the particularities of scheduling procedures in Smart Manufacturing and Industry 4.0 production environments. We reviewed the new technologies associated with these new industrial paradigms and their possible relation with decision-making in production planning, focusing on their integration in CPPS, seen as the main components in these manufacturing environments. We also analysed how these CPPS can handle production plans, in particular choosing the right schedules assigning jobs to machines. We introduced here a new decision-making schema intended to yield flexible and efficient production schedules on the fly, profiting from the features of these new environments. The ability to face unforeseen and disruptive events is one of the main advantages of our proposed Smart Scheduling schema, which using an efficient screening procedure (Tolerance Scheduling) reduces the need of rescheduling in the face of those events.

Future work involves the examination of real-world cases and running Smart Scheduling for them, this will certainly show roads for further improvement of this approach.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Daniel Alejandro Rossit http://orcid.org/0000-0002-2381-4352
Fernando Tohmé http://orcid.org/0000-0003-2988-4519

References

Ahn, G., Y. J. Park, and S. Hur. 2018. "Performance Computation Methods for Composition of Tasks with Multiple Patterns in Cloud Manufacturing." *International Journal of Production Research* 16: 1–14.

Ahuja, R. K., and J. B. Orlin. 2001. "Inverse Optimization." *Operations Research* 49 (5): 771–783.

Al-Hinai, N., and T. Y. ElMekkawy. 2011. "Robust and Stable Flexible Job Shop Scheduling with Random Machine Breakdowns Using a Hybrid Genetic Algorithm." *International Journal of Production Economics* 132 (2): 279–291.

Bożek, A., and M. Wysocki. 2015. "Flexible Job Shop with Continuous Material Flow." *International Journal of Production Research* 53 (4): 1273–1290.

Brucker, P., and N. V. Shakhlevich. 2009. "Inverse Scheduling with Maximum Lateness Objective." *Journal of Scheduling* 12 (5): 475–488.

Brucker, P., and N. V. Shakhlevich. 2011. "Inverse Scheduling: Two-Machine Flow-Shop Problem." *Journal of Scheduling* 14 (3): 239–256.

Dolgui, A., D. Ivanov, S. P. Sethi, and B. Sokolov. 2018. "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*. doi:10.1080/00207543.2018.1442948.

Framinan, J. M., R. Leisten, and R. R. García. 2014. *Manufacturing Scheduling Systems*. London: Springer.

Framinan, J. M., and R. Ruiz. 2010. "Architecture of Manufacturing Scheduling Systems: Literature Review and an Integrated Proposal." *European Journal of Operational Research* 205 (2): 237–246.

Gao, R., L. Wang, R. Teti, D. Dornfeld, S. Kumara, M. Mori, and M. Helu. 2015. "Cloud-Enabled Prognosis for Manufacturing." *CIRP Annals* 64 (2): 749–772.

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. "The Complexity of Flowshop and Jobshop Scheduling." *Mathematics of Operations Research* 1 (2): 117–129.

Hermann, M., T. Pentek, and B. Otto. 2016, January. "Design Principles for Industrie 4.0 Scenarios." In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 3928–3937. IEEE.

Heuberger, C. 2004. "Inverse Combinatorial Optimization: A Survey on Problems, Methods, and Results." *Journal of Combinatorial Optimization* 8 (3): 329–361.

Katragjini, K., E. Vallada, and R. Ruiz. 2013. "Flow Shop Rescheduling Under Different Types of Disruption." *International Journal of Production Research* 51 (3): 780–797.

Koulamas, C. 2005. "Inverse Scheduling with Controllable job Parameters." *International Journal of Services and Operations Management* 1 (1): 35–43.

Lee, E. A. 2008, May. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 363–369. IEEE.

Lee, J., B. Bagheri, and H. A. Kao. 2015. "A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems." *Manufacturing Letters* 3: 18–23.

Li, J. Q., Q. K. Pan, and K. Mao. 2015. “A Discrete Teaching-Learning-Based Optimisation Algorithm for Realistic Flowshop Rescheduling Problems.” *Engineering Applications of Artificial Intelligence* 37: 279–292.

Liu, Y., L. Wang, X. V. Wang, X. Xu, and L. Zhang. 2018. “Scheduling in Cloud Manufacturing: State-of-the-Art and Research Challenges.” *International Journal of Production Research*, 1–26.

McKay, K. N., and J. A. Buzacott. 2000. “The Application of Computerized Production Control Systems in job Shop Environments.” *Computers in Industry* 42 (2-3): 79–97.

Monostori, L. 2014. “Cyber-Physical Production Systems: Roots, Expectations and R&D Challenges.” *Procedia CIRP* 17: 9–13.

Ouelhadj, D., and S. Petrovic. 2009. “A Survey of Dynamic Scheduling in Manufacturing Systems.” *Journal of Scheduling* 12 (4): 417–431.

Pinedo, M. 2012. *Scheduling*. New York: Springer.

Rossit, D., and F. Tohmé. 2018. “Scheduling Research Contributions to Smart Manufacturing.” *Manufacturing Letters* 15 (B): 111–114.

Rossit, D. A., F. Tohmé, and M. Frutos. 2018. “The non-Permutation Flow-Shop Scheduling Problem: a Literature Review.” *Omega* 77: 143–153.

Rossit, D., F. Tohmé, M. Frutos, J. Bard, and D. Broz. 2016. “A non-Permutation Flowshop Scheduling Problem with Lot Streaming: A Mathematical Model.” *International Journal of Industrial Engineering Computations* 7 (3): 507–516.

Tao, F., Y. Cheng, L. Zhang, and A. Y. Nee. 2017. “Advanced Manufacturing Systems: Socialization Characteristics and Trends.” *Journal of Intelligent Manufacturing* 28 (5): 1079–1094.

Vieira, G. E., J. W. Herrmann, and E. Lin. 2003. “Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods.” *Journal of Scheduling* 6 (1): 39–62.

Wang, P., R. X. Gao, and Z. Fan. 2015. “Cloud Computing for Cloud Manufacturing: Benefits and Limitations.” *Journal of Manufacturing Science and Engineering* 137 (4): 040901.

Wang, X., S. K. Ong, and A. Y. C. Nee. 2017. “A Comprehensive Survey of Ubiquitous Manufacturing Research.” *International Journal of Production Research* 1: 1–25.

Wang, L., M. Törngren, and M. Onori. 2015. “Current Status and Advancement of Cyber-Physical Systems in Manufacturing.” *Journal of Manufacturing Systems* 37 (Part 2): 517–527.

Wang, X. V., and L. Wang. 2014. “From Cloud Manufacturing to Cloud Remanufacturing: A Cloud-Based Approach for WEEE Recovery.” *Manufacturing Letters* 2 (4): 91–95.

Wang, L., and X. V. Wang. 2018a. “Latest Advancement in Cloud Technologies.” In *Cloud-Based Cyber-Physical Systems in Manufacturing*, 3–31. Cham: Springer.

Wang, L., and X. V. Wang. 2018b. “Latest Advancement in CPS and IoT Applications.” In *Cloud-Based Cyber-Physical Systems in Manufacturing*, 33–61. Cham: Springer.

Wang, L., and X. V. Wang. 2018c. “Cloud-Enabled Distributed Process Planning.” In *Cloud-Based Cyber-Physical Systems in Manufacturing*, 105–123. Cham: Springer.

Wang, L., X. V. Wang, L. Gao, and J. Váncza. 2014. “A Cloud-Based Approach for WEEE Remanufacturing.” *CIRP Annals* 63 (1): 409–412.

Xu, X. 2012. “From Cloud Computing to Cloud Manufacturing.” *Robotics and Computer-Integrated Manufacturing* 28 (1): 75–86.

Xu, X. 2017. “Machine Tool 4.0 for the new era of Manufacturing.” *The International Journal of Advanced Manufacturing Technology* 92 (5-8): 1893–1900. doi:10.1007/s00170-017-0300-7.