

# Dynamic scheduling of manufacturing systems: a product-driven approach using hyper-heuristics

Wassim Bouazza,  Yves Sallez, Damien Trentesaux 

Lamih, Umr Cnrs 8201, Université Polytechnique Hauts-de-France, Valenciennes, France

## ABSTRACT

Dynamic scheduling of manufacturing systems is encountered in many real-world industries such as the food and pharmaceutical industries. The scheduling of these systems must not only be efficient but also reactive to cope with dynamic job arrivals and machine breakdowns. Over the last decade, products within Product-Driven Control Systems (PDCS) have become smart entities capable of actively handling the manufacturing process. In this paper, a PDCS based on the design of Smart Products is proposed. A generic model of the decisional strategy allows Smart Products to characterize different decisional contexts and thus switch efficiently from one scheduling rule to another using a novel Hyper-Heuristics (HH) based approach. The implementation and testing of the proposed PDCS on hybrid flexible flow-shops with multiple constraints inspired by the pharmaceutical industry are presented. The comparative study with 168 combinations of scheduling rules from the literature highlighted the superiority of the HH to minimize the Mean Completion Time. Furthermore, the proposed approach enhanced both the global performance and the reactivity of the manufacturing control system.

## KEYWORDS

Dynamic scheduling; distributed manufacturing control; product-driven control system; smart product; hyper-heuristics; dispatching rules

## 1. Introduction

Recent changes in market needs such as mass customization and further personalization of products require factories to be smarter than ever (Kim et al. 2020). To handle dynamic scheduling in constantly changing environments, the advances in computing in industry have enabled the advent of distributed architectures to control relevant activities. Indeed, with global trends such as Industry 4.0, Internet of things, or Cyber-Physical Manufacturing Systems, modern factories often host more and more computational entities and decisional nodes that affect global performance. In addition, changes in design and manufacturing processes occur ever more frequently with increasingly complex divisions between production activities and tasks within manufacturing organizations (Chen et al. 2020). Thus, decentralized architectures are more flexible and deal with disturbances better than conventional centralized ones. Nonetheless, these distributed approaches are historically resource-oriented, i.e. the decisional nodes are tied to 'smart' machines that choose the jobs to be performed on 'passive' products (Leitão 2009). Over the last decade, studies have proposed assigning the

decisional abilities to the products to be manufactured (Keddis, Kainz, and Zoitl 2015). Products within Product-Driven Control Systems thus become intelligent entities that can autonomously and, ideally, effectively contribute to the achievement of the different manufacturing stages. In such PDCS, one of the main challenges is defining and improving behavior while facing the risk of inefficient decision-making due to the myopia effect (Blunck et al. 2018).

In that perspective, Hyper-Heuristics seem interesting. Ochoa, Qu, and Burke (2009) define this group of methodologies as 'heuristics to choose heuristics'. Thus, they are research methodologies aimed at automating the process of selecting or combining simple heuristics to solve complex problems. From there, the main idea of this paper is to investigate how a HH can explore and learn the most suitable product behaviors in a dynamic manufacturing system. Indeed, using simulation, the proposed approach can provide the PDCS with a set of behaviors that try to enhance the global performance and reactivity of the manufacturing control system.

The rest of the paper is structured as follows. The second section reviews contributions that address solving the dynamic scheduling of

manufacturing systems. Various existing control architectures are also overviewed with a focus on product-driven approaches. In section 3, a novel HH approach is presented and linked with a PDCS using smart products (SP) based on a generic model of a decisional strategy. Section 4 describes the application to multi-stage manufacturing systems with the specific implementation of a decisional strategy and scheduling rules. Then section 5 details the experiments on two hybrid flexible flow-shops inspired by the pharmaceutical industry. A comparative study with 168 combinations of scheduling rules taken from the literature to minimize the Mean Completion Time is presented and discussed. Finally, the paper is rounded off with a conclusion and a summary of possible future research.

### ***Optimization tools and control architectures for dynamic scheduling of manufacturing systems: a literature review***

Manufacturing environments are far from being static and numerous unpredictable real-time events can occur that perturb the activities and, therefore, scheduling. According to (Vieira, Herrmann, and Lin 2003), events can be classified into two categories:

1. *Job-related events*: in dynamic scheduling, jobs can arrive at some known or unidentified future time. This contrasts with static scheduling in which jobs are known at time 0.

2. *Resource-related events*: these kinds of events concern machine breakdowns, machine unavailability, loading limits, raw material deficiencies, defective materials, etc. For example, when a breakdown occurs, it implies that the machine cannot process anything for a specific duration that can be known or not.

Both types of events frequently imply scheduling adjustments, which is called dynamic scheduling. In the literature, the dynamic scheduling and rescheduling environment, also noted (re)scheduling, involves an unknown (possibly infinite) set of jobs arriving over an unbounded rolling time horizon (Fattahi and Fallahi 2010). Vieira, Herrmann, and Lin (2003) provide a framework to classify the rescheduling approaches. The authors underline that classical static deterministic scheduling can be viewed as a special case of (re)

scheduling: 'a finite set of jobs with no uncertainty about future events or processing duration'.

Dynamic scheduling can be grouped into three categories depending on the type of schedule generated (Ouelhadj and Petrovic 2009). Firstly, in the case of completely reactive scheduling, no detailed schedule is generated in advance. So, a priority rule is used to select the next job among a set of waiting ones by picking the job with the highest priority. Secondly, in predictive-reactive scheduling, the scheduler generates an initial schedule and then applies a scheduling and/or rescheduling process in which the schedules are revised to cope with the occurrence of real-time events. Thirdly, robust pro-active scheduling tries to generate schedules that satisfy performance predictability requirements in a dynamic environment. The main difficulty is then determining the predictability measures.

Over the past few decades, researchers have proposed different optimization approaches and manufacturing control architectures to handle multiple constraints, especially in a dynamic environment. The next section provides an overview of studies relating to dynamic scheduling and the subsequent section reviews control architectures with a focus on product-driven approaches.

#### ***2.1. Optimization tools for dynamic scheduling***

In recent years, many techniques, from simple heuristics to artificial intelligence techniques (e.g. dispatching rules, fuzzy logic, neural networks, multi-agent systems), have been applied to solve dynamic scheduling problems, especially for job shop and flow shop manufacturing plants. The choice of the algorithm applied to the scheduling problem mainly depends on the specificities of the dynamic environment (Kalinowski, Krenczyk, and Grabowik 2013):

(1) *Exact approaches*: As discussed above, a dynamic manufacturing system comprises numerous entities with the occurrence of many real-time events. The resulting interactions between entities lead to emerging properties. Therefore, according to (Framinan, Leisten, and Ruiz García 2014), to be called 'exact' a method fulfilling the requirements of the formal optimization problem must deliver the solution within a finite (possibly large) number of computation steps and in a deterministic manner. Using exact

approaches in the dynamic context addressed is hard to achieve. Indeed, modeling the system as a whole involves models that are more complex than theory can solve (Corning 2002), i.e. the mathematical models are often too complicated to be resolved. Nevertheless, many papers have used mathematical models to calculate a boundary, in particular using relaxation techniques, to make comparisons with more complex methods or to generate a primary schedule that is adjusted later using other algorithms such as meta-heuristics in hybridized approaches. For example, to minimize the total tardiness in a flexible job shop scheduling problem with sequence-dependent setup times, Mousakhani (2013) developed a Mixed Integer Linear Programming mathematical model. First, the model was compared with a mathematical model from the literature. Then, a metaheuristic based on an iterated local search was proposed and compared with Tabu Search and Variable Neighborhood Search algorithms.

2. *Heuristics*: Anuradha and Sumathi (2015) define heuristic as a method that tries to find good solutions at a reasonable computational cost without being able to guarantee optimality. However, in manufacturing scheduling problems, some ambiguity with Dispatching Rules (DR) emerged when reviewing the literature. To avoid any confusion, Trentesaux et al. (2013) distinguished between assignment heuristics (also called allocation rules) and sequencing heuristics. DR are well-known solutions in the literature and are quick and easy to implement (Branke et al. 2016). Nevertheless, they may also provide poor global performance due to real-time local decisions. DR can be significantly enhanced by considering problem specificities (e.g. explicitly taking into account setup times in the decisional process). In (Pickardt and Branke 2012), setup-oriented rules were classified into three categories: purely setup-oriented, the most simple, such as the 'shortest setup time' rule (Gavett 1965), composite rules characterized by a priority index integrating a term to avoid setup operations (Chiang and Fu 2009), and family-based rules characterized by a hierarchical approach. The latter reduce the setup times using family information to combine the processing of jobs belonging to a specific family (Chern and Liu 2003). Thus, it appears that in some cases, DR can produce a fast response by providing a good balance between reactivity and performance. Choi

and You (2006) provide a performance analysis of DR for dynamic scheduling in one-of-a-kind production. The paper also aims to investigate questions such as 'How do we define dynamic dispatching rules?' and 'How do we analyze performance?', and provide guidelines.

It is also interesting to note that methods combining scheduling issues with Game Theory have also attracted attention from researchers, especially in a cloud manufacturing context. Although it does not deal with the sequencing aspect of operations, it proposes an interesting approach for the allocation of tasks to resources. For example, (Zhang et al. 2017) propose a real-time allocation strategy using the Nash equilibrium. The proposed algorithm improves efficiency while reducing the processing cost for a flexible job shop scheduling problem. Another example is provided by Carlucci et al. (2020). In the context of Cloud Manufacturing, the authors propose a noncooperation model based on the Minority Game Theory, especially when few information is provided by competitors (i.e. resources). The simulation results show how the proposed model generates a better solution than a classic formulation taking into account merely experience. The performance obtained is as close to the centralized benchmark as when information is freely available among the participants.

3. *Meta-heuristics*: Henderson, Jacobson, and Johnson (2006) define meta-heuristics as solution methods that orchestrate an interaction between local improvement procedures and higher-level strategies to create a process capable of escaping from local optima and performing a robust search in a solution space. Indeed, exact methods such as Branch & Bound can, under certain conditions, find an optimal solution but at the cost of lengthy computation. Other approaches such as Random Walk or Gradient Search are simple random search and gradient descent approaches that evaluate one solution at a time. In contrast, some meta-heuristics such as Genetic Algorithms (GA) make the search process computationally efficient by simultaneously evaluating multiple solutions (named population or offspring) (Jorapur et al. 2016). For example, to minimize the makespan ( $C_{max}$ ) for a hybrid flow shop scheduling problem with sequence-dependent setup times, Pan et al. (2017) proposed 9 algorithms: three iterated local search variants, three iterated greedy variants, and three population-based

metaheuristics. The results obtained showed that discrete Artificial Bee Colony Optimization was the best-performing algorithm, improving 126 of the 240 solutions referred to as 'best-known solutions' in the literature. In recent years, an increasing number of solutions based on hybridized techniques have appeared, in contrast with pure heuristics or metaheuristics approaches. The idea is to take advantage of the strengths of one or more methods. For example, Liou and Hsieh (2015) proposed a hybrid algorithm for scheduling a multi-stage flow shop. They combined Particle Swarm Optimization and GA considering both sequence-dependent setup and transportation times.

4. *Hyper-Heuristics*: The neologism 'Hyper-Heuristic' was first introduced in a conference paper by Cowling, Kendall, and Soubeiga (2001). Being a group of methodologies for selecting or generating heuristics, several definitions can be found in the literature (Nguyen et al. 2013; Zhang and Roy 2019). However, the common feature that distinguishes hyper-heuristics is that they operate in a search space composed of heuristics. Starting from a set of basic heuristics, they try to select existing ones or generate new ones. Then they apply the appropriate rule considering each decisional point. To classify HHs, a paradigm given by Burke et al. (2010) differentiates between constructive and perturbative searches. Firstly, constructive methods work iteratively starting from partial solutions with one or more missing components that are progressively filled in. Secondly, perturbative methods consider complete candidate solutions and modify one or more of the solution components.

Although the term is recent, the idea is not new. For example, Hershauer and Ebert (1975) developed a standardized approach to select simple sequencing rules for a job shop. The proposed sequencing rule is a linear combination of decision factors, each of which is initially assigned a relative weight. The rule is then used to determine the priority of each job in the queues. Each rule generates a shop cost calculated using simulation. A further application of HH can be found in (Ochoa, Qu, and Burke 2009). As previously mentioned, the authors defined HH as "heuristics that choose heuristics" to solve a given combinatorial optimization problem. Thus, the optimization of

two objective functions using different HH representation sizes (named block sizes) was explored. The study concluded that the proposed heuristic search spaces were suitable for developing solutions to a production scheduling problem. Other papers have explored search spaces composed of heuristics such as DR. For example, Çakar and Cil (2004) used Artificial Neural Networks to design a manufacturing system considered as a composite system combining four different priority rules. The alternatives were evaluated in terms of performance measures and then the best design alternative was selected from the possible alternatives. The proposed system also determines which priority rules can be used in the evaluation and these priority rules directly contribute to measuring system performance. Another example is provided in (Vázquez Rodríguez, Petrovic, and Salhi 2007). A HH was proposed for the optimization of  $C_{max}$  in a hybrid flow shop considering sequence-dependent setup times and uniform parallel machines. The representation used consisted of a sequence of DR to be called one after the other to complete a schedule. Note that Grammar-based representations are also popular. They allow the generation of priority functions of variable format and length (Mascia et al. 2013).

More recently, the term hyper-heuristic has also been used to refer to a learning mechanism for selecting and/or generating heuristics to solve computational search problems (Burke et al. 2013). An additional classification framework of HH approaches was proposed by Burke et al. (2019). The definition of HH was extended by considering them as learning algorithms when they use feedback from the search process. If it exists, this learning can be online when learning takes place while the algorithm solves an instance of a problem or offline when the algorithm gathers knowledge before trying to generalize it later during the execution of the system. An example of this reinforcement feedback is tested in (Bouazza et al. 2015). The authors propose a distributed approach based on intelligent products that use Q-learning to select dispatching rules. The heterarchical control architecture uses simulation to perform Reinforcement Learning and enhance product behavior.

## 2.2. Centralized and Product-Driven Control Systems

Classical solutions based on mathematical analysis and other exact methods have shown their limits in dealing with complex manufacturing environments. Furthermore, the choice of an optimization tool is directly linked to the type of schedule targeted. Another important point is the manufacturing environment and more particularly the information available while the scheduling task is being performed. This is also a crucial point that must be considered for the manufacturing control system architecture.

Various architectures have been proposed to handle the combinatorial problems of scheduling in dynamic manufacturing environments of which the most common are centralized or hierarchical approaches. Entities within centralized architectures broadcast decisions hierarchically from the higher levels down to operational level. In stable industrial environments, they offer good production scheduling optimization (Thomas, Trentesaux, and Valckenaers 2012) but are generally unable to deal with unexpected events. In such cases, when the dynamics of the manufacturing systems vary significantly, heterarchical architectures enabling distributed control are more flexible and provide more agility (Morariu et al. 2014). This shift from centralized to decentralized control, ideally in real-time, offers many possibilities for managing dynamic and complex systems (Windt et al. 2010).

PDCS focus automation on product flow as opposed to conventional resource-oriented approaches that focus on the control of manufacturing cells. Decisions regarding product scheduling and routing can be decentralized. Such product-centric approaches aim to bring intelligence and autonomy as close as possible to the physical system. The decisional entities can then work together to react quickly and autonomously within constraints instead of requesting control decisions from a centralized controller (Borangiu et al. 2014). Based on products endowed with decisional capabilities and using different techniques including agent-based approaches (Akkiraju et al. 2001), holonic architectures (Lei and Yang 2013), or other concepts such as 'active products' (Sallez et al. 2010), PDCS completely transform the products from passive entities to active participants in the decisional process (Herrera and Berraf 2011).

However, studies also exhibit some weaknesses such as 'myopic behavior' that can appear and is caused by local decisions (Zambrano Rey et al. 2014). The fact that each unit inside a heterarchical system has its own objectives and a narrow view of the global system results in selfish behavior affecting overall performance. In (Pannequin, Morel, and Thomas 2009), a benchmark analysis showed that PDCS can perform at least as well as classical centralized control architectures. The authors concluded that the robustness of the solutions depends on local decisions.

To support PDCS, HH appears to be a very interesting solution that can combine fast and responsive heuristics such as DR with an optimization-simulation mechanism. Thus, simulation can be a key tool for exploring and discovering the most suitable decentralized behavior. At this point, optimization via simulation becomes a very promising solution to suppress or at least reduce myopic behavior while keeping the system reactive.

In the next section, an approach based on completely reactive scheduling using HH is proposed. The goal of the developed HH is to provide a solution using both fast scheduling rules and an optimization-simulation mechanism to ensure the good global performance of a manufacturing system.

## 3. Proposed approach

In this section, the key elements of the approach are detailed.

### 3.1. Mathematical formulation

In this paper, the following notations describing the parameters and the variables are used (Table 1).

## 4. Choice and motivations

Considering the discussion in the review section, the contribution must meet the two following requirements:

- *Need for reactivity*: one of the main characteristics of the environment considered is its rapidly changing nature, in particular the unavailability of information on future products before they arrive in the system and unpredictable

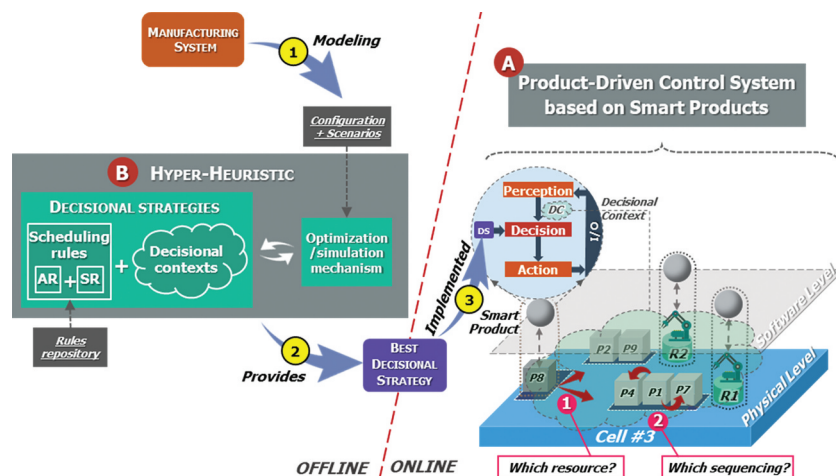
**Table 1.** Parameter and variable notations.

	Notation	Description
Parameters	P	set of products (or jobs), $P = \{1, 2 \dots  P \}$
	F	set of product families, $F = \{F_1, F_2 \dots  F \}$ , $F \subseteq P$
	R	set of resources, $R = \{1, 2 \dots  R \}$
	$f_j$	family of product $j$ , $f_j \in F$ et $j \in P$
	$p_j$	priority of product $j$ , $p_j \in \mathbb{N}^*$ , $j \in P$
	$p_r(f_j)$	processing time of a product from family $f_j$ on resource $r$ , $f_j \in F$ , $r \in R$
	$s_r(f_j)$	setup time of a product from family $f_j$ on resource $r$ , $f_j \in F$ , $r \in R$
	$\delta_r$	set of product families that can be processed by resource $r$ , $\delta_r \subseteq F$
	$b_{rt}$	breakdown duration on resource $r$ , at time $t$ , $b_{rt} \in \mathbb{R}^+$ , $r \in R$
	Variables	$\epsilon_{jr}$
$C_j$		completion time of job $j$ , $C_j \in \mathbb{R}^+$

breakdowns. This makes it almost impossible to apply conventional methods as predictive methods. The solution must be reactive to deal with the numerous disturbances in an uncertain environment.

- *Need for overall performance:* in addition to the need for responsiveness, the solution must also be effective. This need, however, involves controlling the emerging behaviors of complex and dynamic interactions.

To meet these requirements, our proposal is based on the integration of two pillars depicted in Figure 1. The synergy resulting from their integration makes it possible to build a global solution consistent with the main requirements:



**Figure 1.** Global view of the proposed approach.

(A) PDCS based on smart products (Figure 1-A): the highly dynamic and uncertain environment addressed has encouraged reactive management. The latter is supported by a purely distributed control architecture based on SPs, providing reactivity, flexibility, and agility. In the proposed approach, products make decisions and resources are considered as passive entities with no decisional capabilities. This product intelligence can be embedded using microcontrollers or other electronic devices with transmission, storage, and information processing capabilities (e.g. using SoC, RFID, ZigBee), as well as remote, i.e. the decisional entity does not physically follow the product around the shop floor but controls it remotely (e.g. a distant entity running on an industrial application server). The SP can then act autonomously in the manufacturing system to undergo manufacturing operations depending on its manufacturing process. However, this choice of distributing the decisional nodes induces another issue, namely myopia control. Indeed, this effect must be considered to limit its negative impact on overall performance.

(B) Hyper-heuristic (Figure 1-B): a HH is proposed to provide the SP in the manufacturing system with efficient behavior. As highlighted in Figure 1, the HH is based on the coupling of decisional strategies with an optimization-simulation mechanism. Indeed, to increase the reactivity of the PDCS, the heart of the solution is based on scheduling rules. In addition to their relative simplicity, they can be applied very

quickly in numerous situations. Nevertheless, their global performance is often poor. To lift this important limitation and improve global performance, one solution may lie in combining several rules. As illustrated by the example in Figure 2, it may be interesting, under certain conditions, to combine different rules by switching from one to the other at the appropriate time.

The decisional strategies are then built by combining a set of both sequencing and allocation rules. These candidate rules, called “atomic rules”, are implemented and grouped in a structure called “rules repository”. In this perspective, the concept of “Decisional Context” (DC) that is suggested makes it possible to select the rules according to the status of one or more significant indicators (e.g. size of the waiting queue above or below a certain threshold) (Bouazza et al. 2019). The DC allows the state of the manufacturing environment to be characterized at a given moment to determine which rule is the most suitable to be applied. The idea of the overall process is illustrated in Figure 1 (yellow numbered circles), and unfolds as follows:

- (1) Starting from a given manufacturing system, modeling allows both its characteristics through the shop floor configuration and dynamic scenarios with product arrivals and possible disturbances to be described.
- (2) Using a set of rules within the rules repository, the HH starts an optimization-simulation mechanism. Once the optimization via simulation is completed, a unique decisional strategy is generated that is intended to be applied by all SPs. This mechanism is not detailed in this paper, but the application section

contains an example implemented using a genetic algorithm.

- (3) Finally, the best decisional strategy thus obtained can be implemented in SPs in the real manufacturing system. However, this work does not concern the last phase of online implementation, which has only been validated through simulation for the time being as well.

The next two sections provide a detailed description of the main components of the proposed approach.

#### 4.1. Control system based on smart products

As illustrated in Figure 1, to make the SP autonomous and responsive, an internal architecture is proposed.

The architecture comprises four modules as follows:

- *I/O module*: this communication module manages interactions between the SP and its environment.
- *Perception module*: processes incoming data to identify the current decisional context.
- *Decision module*: considering the current DC, and using its decisional strategy, the SP chooses the appropriate scheduling rule.
- *Action module*: translates the rules selected by the decision module into a set of actions.

Figure 1 (numbered circles on the right) and Figure 3 highlight the decisional process applied by the SP:

- (1) After collecting and processing data relating to the state of the manufacturing system to determine the current DC, the SP applies the allocation rule to select a resource from the set of candidates.

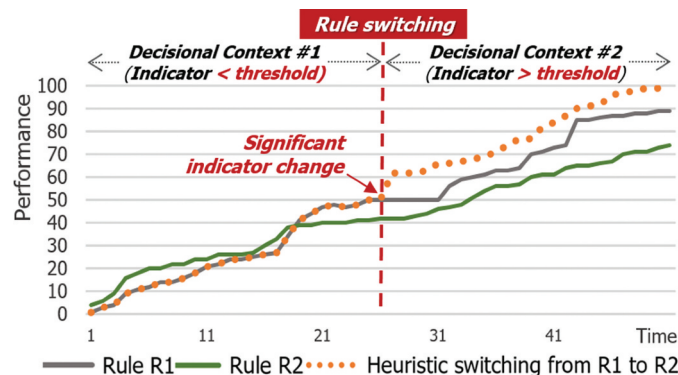


Figure 2. Performance of two rules vs rule switching.

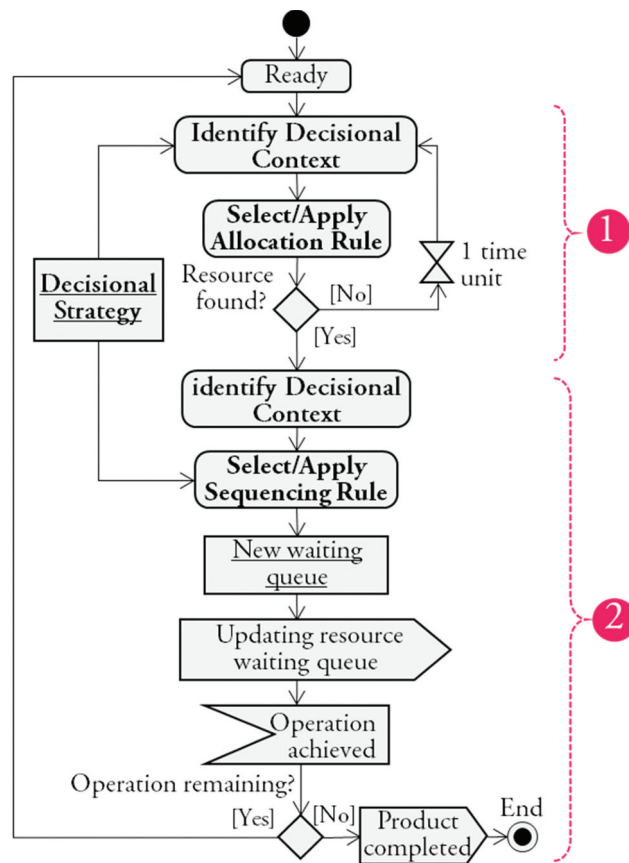


Figure 3. Activity diagram of the smart product.

(2) After joining the waiting queue of the selected resource, as with the previous allocation rule, the sequencing rule corresponding to the new DC is selected then applied. The new waiting queue (i.e. a sequence of operations corresponding to the different products waiting to be processed on the resource) is configured on the resource. The SP then waits until the end of the required operation. Once the operation is completed, if there are still operations to be performed, the SP repeats the decisional process. Otherwise, the SP is considered finished.

The next section describes the generic decisional strategy model embedded in the SP.

#### 4.2. Hyper-heuristic proposed

As illustrated in Figure 4, the decisional strategy is composed of two policies, one for the resource allocation sub-problem and one for the sequencing sub-problem. Each policy comprises a set of heuristics. Each heuristic is in turn composed of one or more scheduling rules. Therefore, discriminating variables

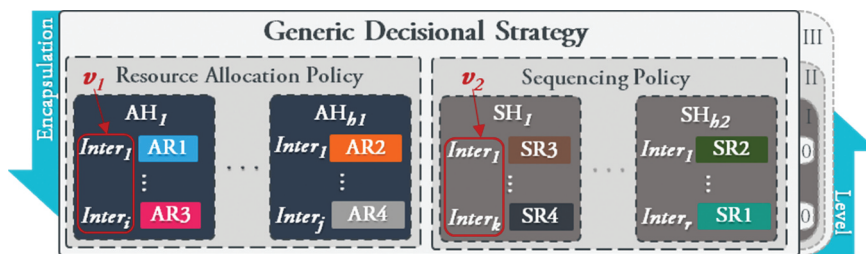


Figure 4. Generic model of the proposed decisional strategy.



are used to characterize the different DC conditioning the selection of one rule or another. To formalize the proposed generic model of the decisional strategy based on intervals, two parameters and two variables are defined:

$h_1$	number of allocation heuristics in the resource allocation policy
$v_1$	discriminating variable for selecting allocation heuristics
$h_2$	number of sequencing heuristics in the sequencing policy
$v_2$	discriminating variable for selecting sequencing heuristics

Four levels within the decisional strategy are distinguished as follows:

**Level-0:** this is the lowest level, comprising the so-called ‘atomic’ scheduling rules.

- **Allocation Rule (AR):** allows the SP to select a resource from a set of candidate machines (e.g. LQE for ‘Least Queued Elements’).
- **Sequencing Rule (SR):** is used to define the sequence of operations on a given resource by sorting its waiting queue (e.g. FIFO for ‘First in First Out’).

**Level-I:** this level is composed of a set of rules to which intervals are added.

- **Allocation Heuristic (AH):** composed of a set of AR with two bounds. These intervals correspond to the ranges of values of the discriminating variable  $v_1$  conditioning the use of an allocation rule (e.g. if  $v_1 \in ]0,5]$  use AR1, else if  $v_1 \in ]6,11]$  use AR4, etc.).
- **Sequencing Heuristic (SH):** similar to AH, it is composed of one or more SR plus intervals of values that condition its triggering (e.g. if  $v_2 \in ]0,7]$  use SR2, else if  $v_2 \in ]7,11]$  use SR3, etc.).

The intervals must be contiguous to cover the set of values that the discriminating variable can take, and this for both AH and SH.

**Level-II:** this level is used to group the different heuristics according to the scheduling sub-problem handled by each policy.

- **Resource Allocation Policy:** composed of  $h_1$  AH. Depending on the context (e.g. the current cell), the corresponding AH is called (see ‘ITV’ in the next section).

- **Sequencing Policy:** composed of  $h_2$  SH, it is also triggered depending on the context (e.g. current resource).

**Level-III:** this is the highest level of encapsulation. It represents the entire decisional strategy comprising the two policies.

This model makes it possible to encapsulate the scheduling rules and the DC that will condition their triggering by the SP. It is also intended to be as generic as possible and independent of the production system configuration as well as the scheduling rules implemented in the rules repository. In this way, the HH goes through a search space composed of instances of decisional strategies. The aim is to produce the most efficient one possible.

The next section details the application to multi-stage manufacturing systems with constraints inspired by the pharmaceutical industry.

## 5 . Application to multi-stage manufacturing systems

First, the context of the study and the multi-stage manufacturing systems addressed are presented, then the implementation of the hyper-heuristic through a specific decisional strategy model is described.

### 5.1. Context of the study

This case study was inspired by the pharmaceutical industry. Indeed, some characteristics frequently encountered in this sector such as multiple product families and heterogeneity of resources are highlighted. In addition, this type of production system is interesting in terms of flexibility and complexity, and exhibits the following specific characteristics:

(1) *Variety of processes:* an essential factor is the flexibility of the manufacturing system that can result from the variety of products and/or resources.

- (a) *Variety due to products:* although production lines are most often dedicated to a specific pharmaceutical class (e.g. analgesic, anti-inflammatory, antibiotic), multiple product families can coexist on the shop floor. It is not unusual that the same unit can produce

different drug molecules (e.g. ibuprofen, keto-profen) and the same molecule can be produced in different galenic forms (e.g. tablet, capsule, injectable). These particularities have a strong impact on the number and order of operations constituting the manufacturing processes.

- (b) *Variety due to resources*: due to the rapid evolution of manufacturing processes but also production tools, heterogeneous machines using very different technologies can be found within the same manufacturing cell. If there is any incompatibility between at least one resource and a product family, then the cell is partially flexible, otherwise, it is fully flexible.
- (c) *Process variety*: This diversity in terms of both quantity and nature of the products implies variations in processing times and setup times. These setup operations tend to be frequent and mandatory (e.g. calibration, tool change, cleaning and/or sterilization operations). In addition, these operations can be particularly time-consuming.

2. *Variation in production orders*: pharmaceutical production is subject to numerous variations in production orders.

- (a) *Market fluctuations*: uncertainties for example linked to shortages, epidemics, natural disasters, etc.
- (b) *Complex composition of the products and their perishable nature*: this can lead to disruptions in the supply chain.

**Table 2.** Notations of extra parameters and variables.

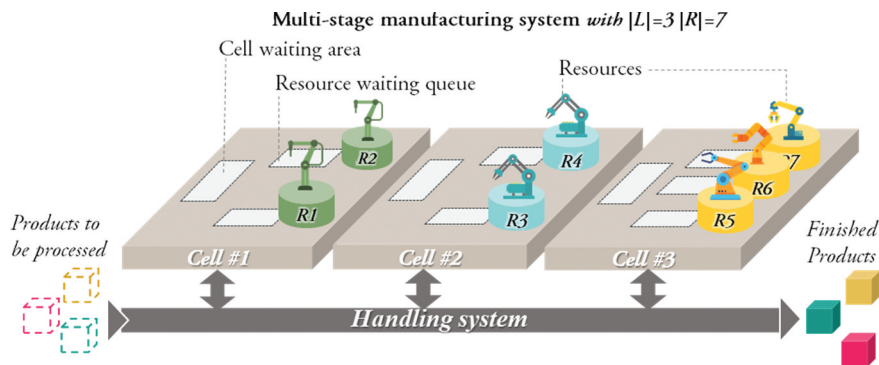
	Notation	Description
Parameters	$L$	set of cells, $L = \{L_1, L_2, \dots, L_{ L }\}$ , $L \subseteq R$
	$I_j$	ordered set of operations for product $j$ , $I_j = \{1, 2, \dots,  I_j \}$ , $j \in P$
variables	$O_{ij}$	operation $i$ of product $j$ , $i \in I_j$ , $j \in P$
	$a_{ij}$	arrival of product $j$ in cell for operation $i$ , $j \in P$
	$w_{ij}$	waiting time before starting setup or processing operation $O_{ij}$ of product $j$ , $w_{ij} \in \mathbb{N}$
	$p_{ijr}$	processing time of operation $O_{ij}$ on resource $r$ , $p_{ijr} \in \mathbb{N}$
	$s_{ijr}$	setup time of operation $O_{ij}$ on resource $r$ , $s_{ijr} \in \mathbb{N}$

- (c) *Technical and regulatory constraints*: indeed, production is subject to the approval of internal quality assurance departments and control bodies of health authorities. These variations can, among other things, lead to unexpected arrivals of product orders due, for example, to authorizations sometimes issued progressively over time.

3. *Breakdowns*: finally, relatively frequent breakdowns must also be considered. These perturbations can also involve safety shutdowns to avoid any risks regarding product quality.

### 5.2. Manufacturing systems and performance indicators retained

The multi-stage manufacturing systems to which the proposed method has been applied, as well as the indicators used to measure the performance, are presented here. For that purpose, extra parameters and variables have been added (Table 2).



**Figure 5.** Example of a multi-stage manufacturing system.

As depicted in the example in Figure 5, three main components of the multi-stage manufacturing system are considered:  $|L|$  cells,  $|R|$  resources, and  $|P|$  products. At each stage, a manufacturing cell groups one or more resources according to their similarities (i.e. operations they can process). Each cell has a waiting area containing products not yet allocated to a resource. Moreover, each resource has a specific waiting queue. The waiting time  $w_{ij}$  of each operation  $O_{ij}$  corresponds to the total time the products spend in these waiting areas, i.e. between the arrival in the cell  $a_{ij}$  and the start of the processing operation (or setup if needed).

The two most commonly used means of measuring performance in the scheduling literature are Makespan ( $C_{max}$ ) and Mean Completion Time (MCT). However, for dynamic scheduling problems with different product arrivals, focusing on the MCT is more suitable. Indeed, in many industries such as electronics or pharmaceuticals, minimizing the MCT may help reduce inventory, holding costs, and/or increase customer satisfaction (Aydivlek and Allahverdi 2012).

Thus, for a product  $j$  starting its process on resource  $r$  and ending on resource  $r'$ , the completion time is calculated as follows:

$$C_j = a_{1j} + (W_{1j} + (\epsilon_{jr} \times S_{1jr}) + p_{1jr}) + (W_{|j|j} + (\epsilon_{jr} \times S_{|j|jr}) + p_{|j|jr}) \quad (1)$$

According to equation (1), the completion time of each product  $j$  corresponds to the sum of the arrival time in the first cell ( $a_{1j}$ ) plus, for each operation (i.e. from 1 to  $|j|$ ), the waiting time, setup time (if any), and processing time.

Then:

$$\sum C_j \text{ is the total completion time, } \sum C_j \in \mathbb{N} \quad (2)$$

$$\sum C_j / |P| \text{ is the Mean Completion Time (MCT), } \sum C_j / |P| \in \mathbb{R} \quad (3)$$

To illustrate the mathematical notations used, Figure 6 provides an example of a Gantt diagram of the scheduling of 3 products showing the main parameters and variables.

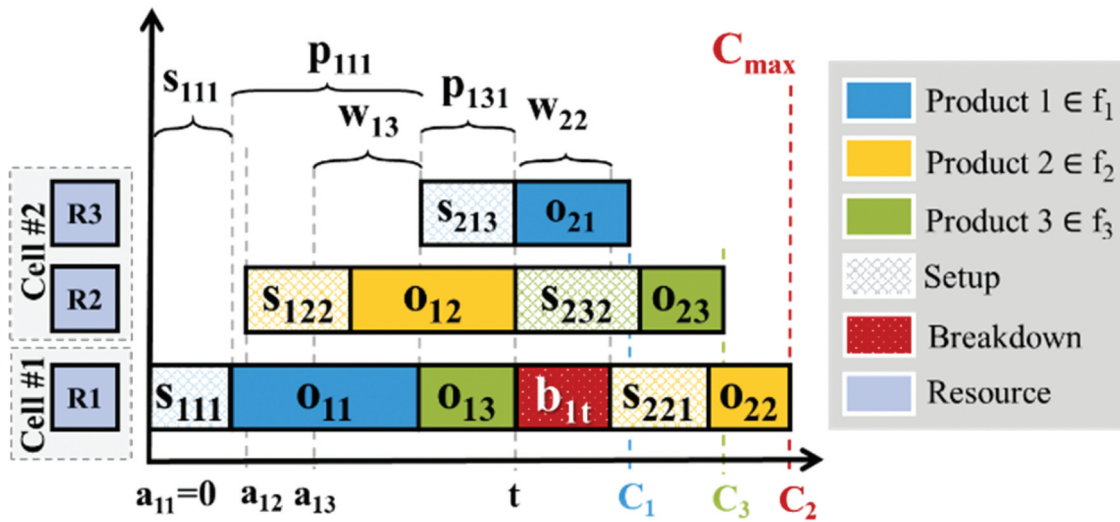


Figure 6. Example of a Gantt diagram of a scheduling.

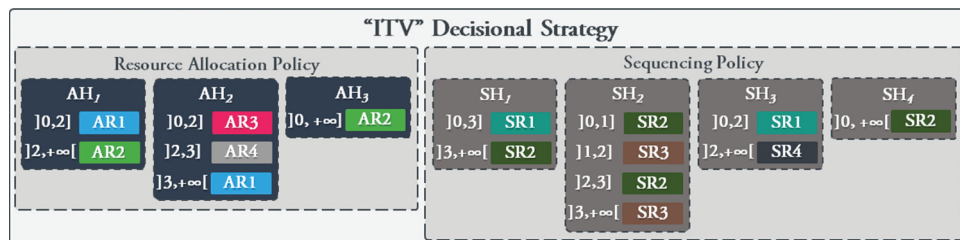


Figure 7. Example of rule encapsulation for the proposed 'ITV' strategy.

**Table 3.** Basic rules repository.

Allocation Rules			Sequencing Rules		
#	Name	Signification	#	Name	Signification
1	LQE	Least Queued Elements	1	FIFO	First In First Out
2	MQE	Most Queued Elements	2	LIFO	Last In First Out
3	STPT	Shortest Total PT <sup>a</sup>	3	FASFO	First At Stage First Out
4	LTPT	Longest Total PT	4	LASFO	Last At Stage First Out
5	STST	Shortest Total ST <sup>b</sup>	5	HPF	Highest Priority First
6	LTST	Longest Total ST	6	LPF	Lowest Priority First
7	QTG	Quickest Total Gross	7	SJF	Shortest Job First
8	LTG	Longest Total Gross	8	LJF	Longest Job First
9	QPT	Quickest PT	9	SST	Shortest ST first
10	LPT	Longest PT	10	LST	Longest ST first
11	QST	Quickest ST	11	SGOS	Shortest Gross first
12	LST	Longest ST	12	LGOS	Longest Gross first
13	QG	Quickest Gross			
14	LG	Longest Gross			

<sup>a</sup>Processing Time<sup>b</sup>Setup Time

### 5.3. Implementation of the hyper-heuristic

To minimize the MCT in a multi-stage manufacturing system, the proposed implementation of the decisional strategy model is detailed, as well as a list of the different atomic rules composing the rules repository.

#### 5.3.1. 'ITV' strategy implemented

As depicted in the example in Figure 7, the 'ITV' strategy (so named in reference to its structure using intervals) makes it possible to describe the behavior of the SP throughout the different stages of a manufacturing system via several decisional contexts. The heuristics are applied as follows: an AH for each cell ( $h_1 = |L|$ ) and an SH for each resource ( $h_2 = |R|$ ). For the discriminating variables, we consider:

- For AH, the number of products in the cell, so  $v_1 \in \mathbb{N}^*$
- For SH, the number of products in the resource waiting queue, so  $v_2 \in \mathbb{N}^*$

Thus, the SP can adapt its behavior not only according to the current cell and resource but also according to the number of products in the different waiting areas and queues.

#### 5.3.2. Atomic rules implemented

The HH builds strategies using the atomic rules listed in Table 3. These 26 rules (i.e. 14 AR and 12 SR) were taken from the literature e.g. (Panwalkar and Iskander 1977; Shahzad and Mebarki 2016) and combined to serve as a basis for comparison of HH performance.

## 6. Experiments

The experiments aim to provide a comparative study of the performances of PDCS using HH with ITV strategies compared to PDCS in which the SPs use a combination of just two scheduling rules (i.e. AR+SR).

In this section, the experimental protocol, the tools, and the key parameters used to generate the different instances are presented. The results of the 24 different scenarios are then summarized and discussed.

### 6.1. Experimental protocol

To evaluate the proposed approach and its implementation, it is necessary to study the different behaviors of the SPs composing the PDCS under different conditions. The problem instances in this paper were inspired by the pharmaceutical industry. A set of tools was specifically developed to model different configurations enabling the use of dynamic manufacturing scenarios with various unexpected events. For each scenario, 168 executions (i.e. 14 AR x 12 SR) of atomic rule combinations were tested to determine the Best Rules Combination (BRC) each time. Additionally, as the HH ITV is a non-deterministic algorithm (due to GA), 30 independent runs were conducted for each scenario. This number is considered to be statistically sufficient in many studies, e.g. (Pérez-Rodríguez and Hernández-Aguirre 2018).

Furthermore, two-way Analysis Of Variance (ANOVA) was systematically conducted for each configuration to set up the HH in the best possible way. ANOVA considers two main parameters of the GA: the size of the population 'Pop' and the replacement rate 'Rate'. These two factors can have 5 values: 100, 250, 500, 750, and 1000 for 'Pop', and 98%, 95%, 90%, 80%, and 50% for 'Rate'. The 25 possible combinations of factors were evaluated 30 times for each configuration.

### 6.2. Generation of problem instances

As illustrated in Figure 8, each instance is composed of two elements: the first 'static' part concerns the manufacturing system configuration (cells, resources, processing and setup times, etc.). The second 'dynamic' part consists of a scenario describing the different events (product arrival rates, breakdowns, etc.).

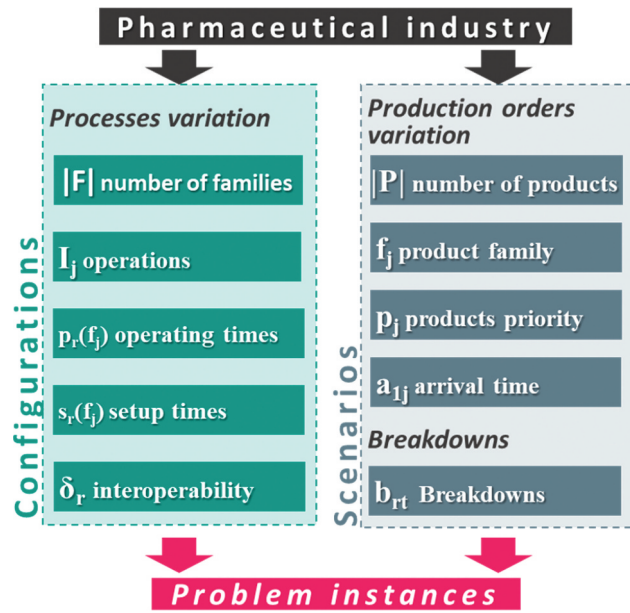


Figure 8. Control of experimental parameters for instance generation.

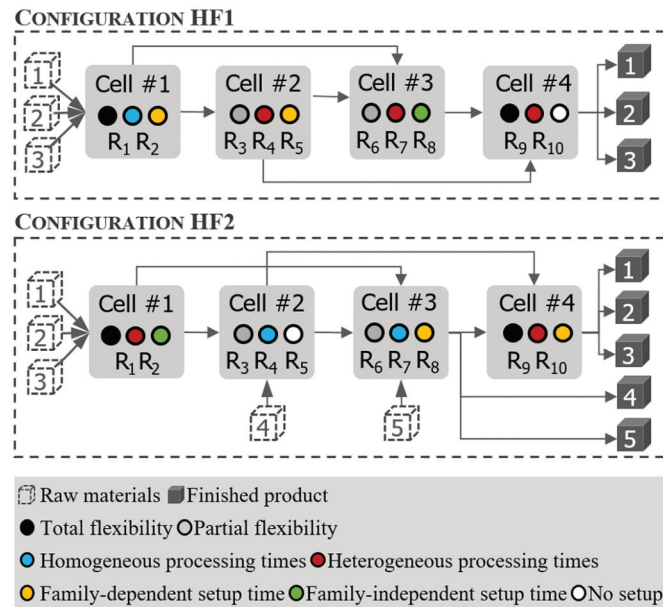


Figure 9. Hybrid flexible flow shop configurations HF1 and HF2.

### 6.2.1. Configurations of multi-stage manufacturing systems

As exhibited in Figure 9, two multi-stage manufacturing systems were retained. These are two hybrid flexible flow shops named HF1 and HF2. These types of

configurations are frequently encountered in real-world industries but remain generic enough to investigate the validity of the proposal. These configurations allow different levels of ‘routing’ flexibility (i.e. the ability to use alternative paths within the

**Table 4.** Precedence constraints according to product families.

Configuration F	amily	Operation Sequence			
#1	F <sub>1</sub>	O <sub>1j</sub>	O <sub>2j</sub>	O <sub>3j</sub>	O <sub>4j</sub>
	F <sub>2</sub>	O <sub>1j</sub>	O <sub>3j</sub>	O <sub>4j</sub>	
	F <sub>3</sub>	O <sub>1j</sub>	O <sub>2j</sub>	O <sub>4j</sub>	
#2	F <sub>1</sub>	O <sub>1j</sub>	O <sub>2j</sub>	O <sub>3j</sub>	O <sub>4j</sub>
	F <sub>2</sub>	O <sub>1j</sub>	O <sub>3j</sub>	O <sub>4j</sub>	
	F <sub>3</sub>	O <sub>1j</sub>	O <sub>2j</sub>	O <sub>4j</sub>	
	F <sub>4</sub>	O <sub>2j</sub>	O <sub>3j</sub>	O <sub>4j</sub>	
	F <sub>5</sub>	O <sub>3j</sub>	O <sub>4j</sub>		

manufacturing system) and ‘machine’ flexibility (i.e. switching from one operation to another by a machine without requiring excessive effort in terms of time, cost, or performance). Furthermore, the distinction is made between two cell types: ones with homogeneous processing times, i.e. the same processing time regardless of the machine selected by the product, and heterogeneous cells where this time varies from one resource to another.

For both configurations  $|R| = 10$  and  $|L| = 4$ , which corresponds to 10 resources grouped in 4 cells:

(1) *Hybrid flexible flow shop HF1*: in this configuration, the first and the last cells comprise 2 machines, whereas cells #2 and #3 comprise 3 machines. Cells #1 and #4 are fully flexible whereas the other two cells are partially flexible. The processing times are heterogeneous, except for the first cell where they are identical regardless of the machine. Setup times vary by product family for the first two cells but vary by resource for the third and are zero for the last cell.

2. *Hybrid flexible flow shop HF2*: as in the HF1 configuration, the first and last cells comprise 2 machines, whereas cells #2 and #3 comprise 3 machines. Similarly, cells #1 and #4 are fully flexible whereas cells #2 and #3 are only partially flexible. The processing times are

homogeneous for cells #2 and #3, and heterogeneous for cells #1 and #4. The setup times are independent of families for the first cell, zero for the second cell, and dependent on product families for cells #3 and #4.

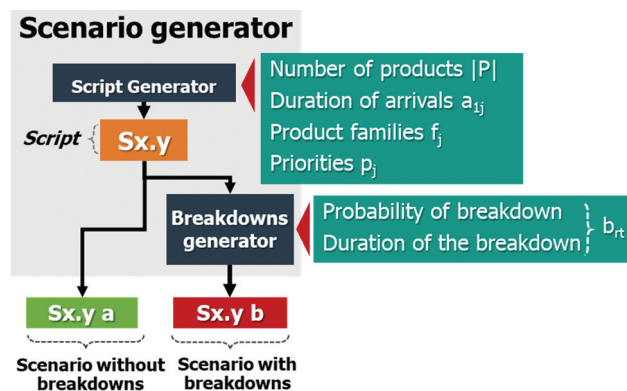
Each product family has a set of operations to be carried out in a well-defined order. Configuration HF1 has  $|F| = 3$  product families, whereas HF2 is a bit more complex with  $|F| = 5$  product families. The precedence constraints defining the number and order of operations  $O_{ij}$  of each family are described in Table 4.

### 6.2.2. Scenario generator

To generate manufacturing scenarios for each configuration, a tool developed using stochastic models allowed parameters such as the number of products, their arrival rates, priorities, and families, as well as the number and duration of disturbances (i.e. machine breakdowns), to be varied. All the scenarios generated are available online.<sup>1</sup> The files in CSV format describe the dynamic events in discrete-event simulations of a manufacturing system with both dynamic job arrivals and breakdowns. Figure 10 shows the scenario generation process:

(1) First, files called scripts, describe  $|P|$  products (identifier, family  $f_j$ , and priority  $p_j$ ) and their arrival times,  $a_{1j}$ . Each script is named in the format ‘x.y’ where x represents the manufacturing system configuration (1 or 2) and y the script number (from 1 to 6). Figure 11 shows the different product arrival rates per script.

2. Secondly, each script allows two scenarios to be generated:



**Figure 10.** Scenario generation process.

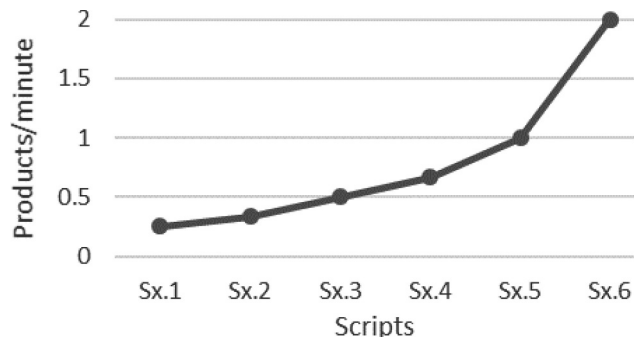
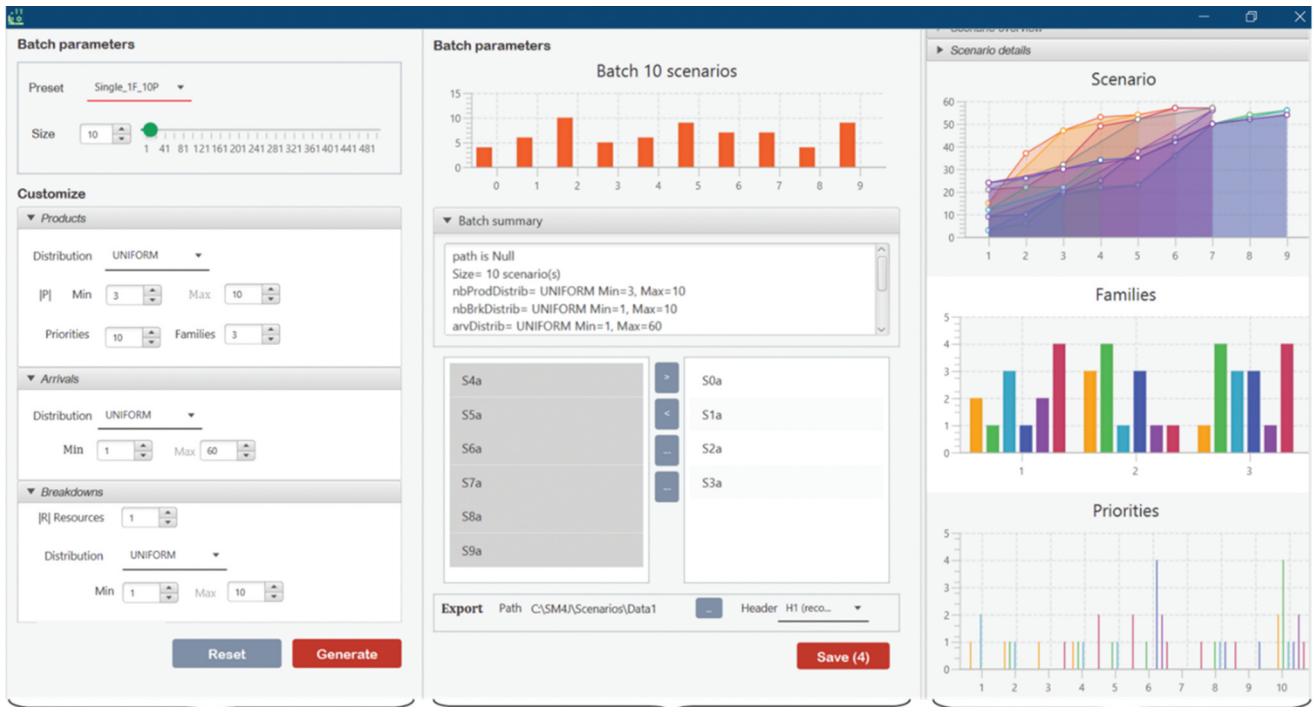


Figure 11. Product arrival rates per script.



Batch parameters: size, products arrivals, breakdowns, etc.

Scenario selection and export within the batch obtained

Graphical visualization of the scenarios generated: arrivals, families and priorities

Figure 12. GUI of the probabilistic tool developed for scenario generation (with Java/JavaFX).

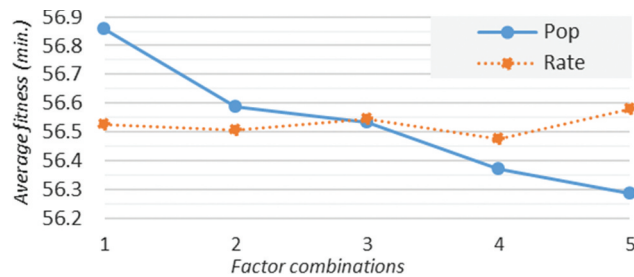


Figure 13. Plot of main effects (data means) of ITV on HF1.

- (a) A first scenario, indicated by the suffix 'a', takes the script as is, with no perturbation.
- (b) The second scenario, indicated by the suffix 'b', takes the script and adds a certain number of disturbances. The latter are generated according to two parameters: the probability of occurrence and the duration of the disturbance determined at random.

Thus, for each configuration, the 6 scripts give rise to 12 scenarios. The product family distribution determines

**Table 5.** Results of the two-way ANOVA on HF1.

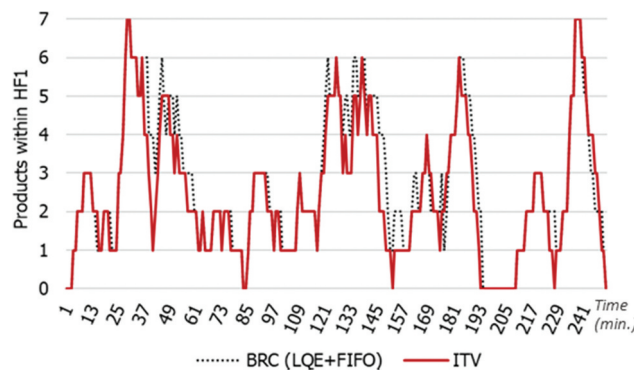
Source of variation	Sum of squares	Degrees of freedom	Mean of squares	F <sub>calc</sub>	Prob.
Pop	4.9148	4	1.2287	22	4.7777E-13
Rate	0.1610	4	0.0402	0.7210	0.5795
Pop*Rate	0.9552	16	0.0597	1.0689	0.3944

the number of operations for each scenario. Each product may require between 2 and 4 operations (see

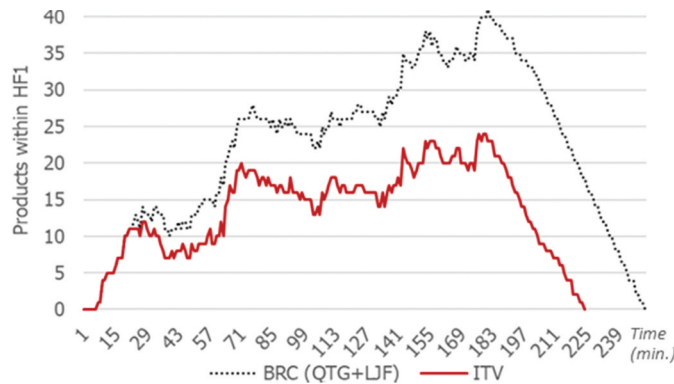
Table 4). So, each scenario requires between 177 and 410 operations, which is quite a significant number compared to the literature but still realistic in industry. The results obtained through simulations are given in the following section.

### 6.3. Experimental results

Regarding the simulation tool, the proposed approach, which is based on a distributed architecture, conditioned the choice of the multi-agent platform. Simulations were conducted with specific tools (Figure 12) developed using Java Eclipse IDE and a JADE (Java Agent DEvelopment framework) multi-agent platform. The simulations were performed on a workstation with an Intel® Xeon® E3-1230 v3 @3.30 GHz CPU and 8GB of RAM. For each configuration, the simulations required between 60 and 120 minutes of computing time depending on the complexity of the configuration and the length of the scenarios. These relatively low execution times



**Figure 14.** Products in HF1 over time for scenario S1.1a.



**Figure 15.** Products in HF1 over time for scenario S1.4b.



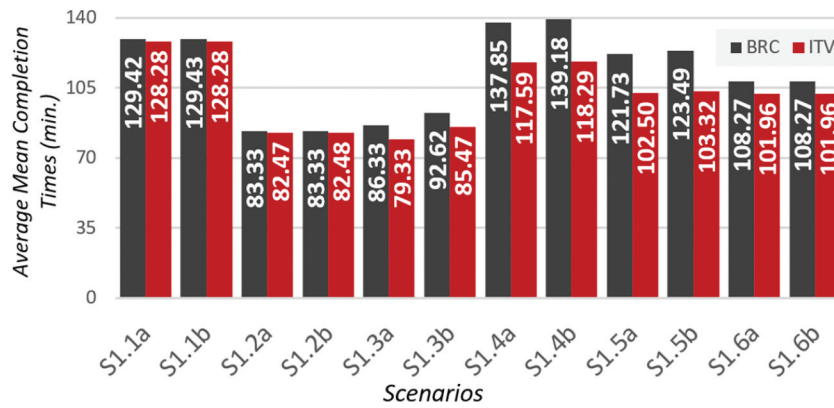


Figure 16. Mean completion times for HF1.

Table 6. Results of the two-way ANOVA on HF2.

Source of variation	Sum of squares mean	Degrees of freedom	Mean of squares	F <sub>calc</sub>	Prob.
Pop	44.6935	4	11.1733	14.9304	1.3371E-09
Rate	1.9945	4	0.4986	0.6662	0.6168
Pop*Rate	9.9512	16	0.6219	0.8310	0.6479

were made possible by the parallelization of the multi-agent containers.

### 6.3.1. Configuration HF1

For the HH ITV, the two-way ANOVA (Figure 13) shows that the size of the population 'Pop' has a statistically significant impact with a p-value < 0.05 (Table 5). The replacement rate has a slight impact on the performance.

Considering these ANOVA results, the parameters used to set up the GA of the HH are Pop = 1000 and Rate = 80%.

Figure 14 shows the evolution of the number of products in the manufacturing system for scenario S1.1a. For this first scenario, the best combinations consisted of the LQE rule associated with FIFO. Using this

combination, the 60 products were produced with an MCT of 129.42 min. The HH ITV lowered the MCT to 128.28 min.

Another example is depicted in Figure 15. For scenario S1.4b with 10 resource breakdowns, the best rule combination was QTG+LJF with an MCT of 139.18 min. The HH ITV lowered it significantly to 118.29 min.

An important aspect is shown in Figure 16, which presents the relative performance of ITV compared to BRC for each of the 12 scenarios in HF1. It appears that ITV still performs better than BRC. This gain varies according to the scenarios.

### 6.3.2. Configuration HF2

The two-way ANOVA results presented in Table 6 and Figure 17 show that the Pop factor has a statistically significant impact in contrast to the replacement rate.

Taking into account the above results, the parameters used to set up the HH are Pop = 1000 and Rate = 90%.

Figure 18 shows the trends in the amounts of products for scenario S2.2a for which the BRC was LPT+LIFO with an MCT of 106.23 min. ITV was more efficient with an MCT of 104.67 min.

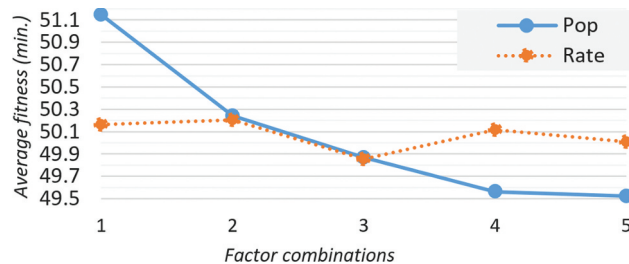


Figure 17. Plot of main effects (data means) of ITV on HF2.

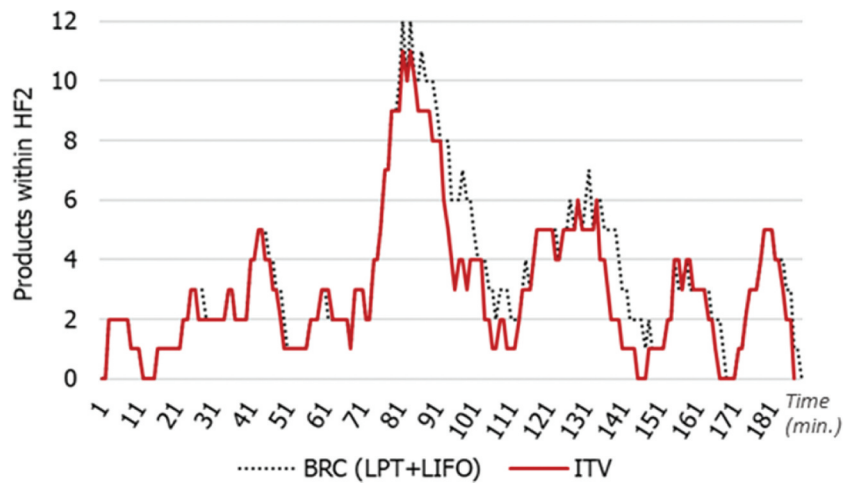


Figure 18. Products in HF2 over time for scenario S2.2a.

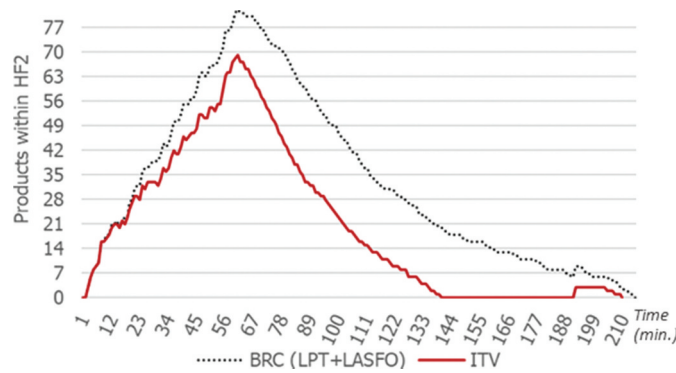


Figure 19. Products in HF2 over time for scenario S2.6b.

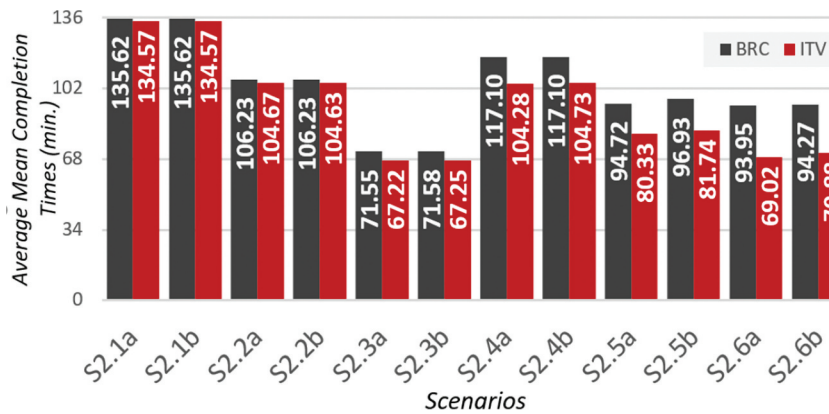


Figure 20. Mean completion times for HF2.

One last example is provided in Figure 19. Scenario S2.6b includes 3 breakdowns. Compared to S2.6a, the LPT allocation rule is in both BRCs. However, the sequencing rules are LST and LASFO for S2.6a and S2.6b, respectively. ITV provides an MCT of 70.88 min, whereas the BRC LPT+LASFO provides a considerably higher MCT of 94.26 min.

Figure 20 shows the average MCTs for the 12 scenarios in HF2. As with HF1, the HH outperforms each time the 168 combinations of atomic scheduling rules.

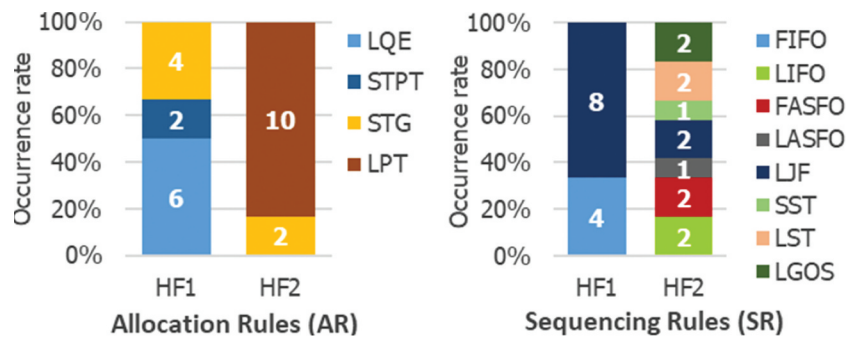


Figure 21. Occurrences of both Allocation and Sequencing Rules in the BRCs per configuration.

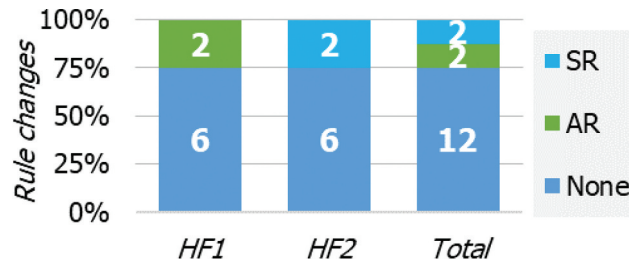


Figure 22. Rule changes in the BRC for scenarios with and without breakdowns.

## 6.4. Discussion

The purpose of this section is to discuss the results of the simulations presented above. Two main aspects are considered. The first concerns the pertinence of rule switching. The second aspect concerns the performance of the proposed HH.

### 6.4.1. Study of the rule switching

The main focus here is on the impact the configurations, as well as breakdowns, may have on the BRCs.

(1) *Impact of configurations*: by observing the BRCs in the two manufacturing system configurations (Figure 21), it appears that only 4 AR out of the 14 rules implemented appear in the BRCs: 3 rules for HF1 and 2 rules for HF2. The results for SR are slightly

different; 8 SR among the 12 rules implemented appear in the 24 BRCs: 2 SR for HF1 and 7 SR for HF2. This reveals that the performance of the rules is closely linked to the problem instances, that is to say, both configurations and scenarios.

2. *Incidence of breakdowns*: with identical scripts (i.e. same configuration and same product arrivals), the occurrence of breakdowns may induce a change in the BRC. Thus, a combination of rules that perform best for a scenario without disturbances is overridden by another combination in the presence of breakdowns. Figure 22 shows rule changes by comparing the BRCs two by two. Rule changes appear in a third of cases. These results show that the need to switch from one rule to another also depends on whether breakdowns occur or not.

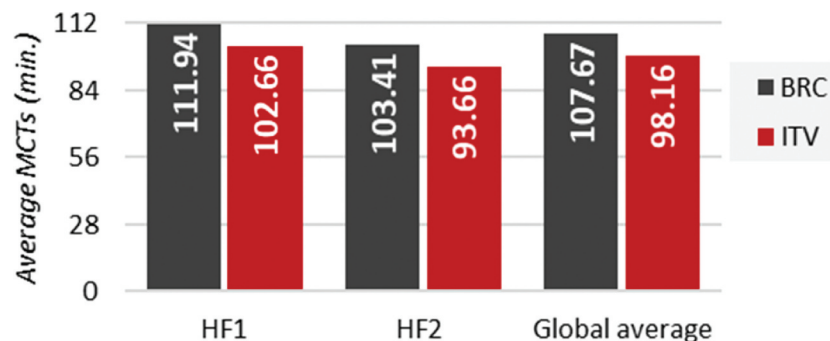


Figure 23. Average Mean Completion Times (MCTs).

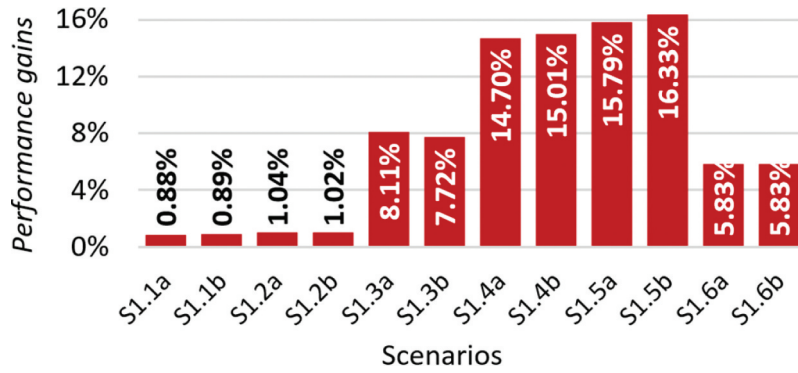


Figure 24. Performance gains with ITV compared to BRC in HF1.

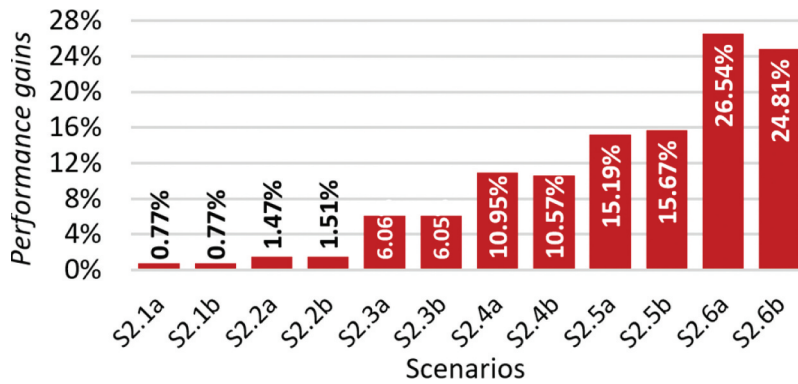


Figure 25. Performance gains with ITV compared to BRC in HF2.

### 6.4.2. Study of HH performance

This section discusses the performance of the HH from three different angles: instances (i.e. both configurations and breakdowns), product arrival rates, and finally, the convergence of the HH.

(1) *Impact of instances:* as shown in Figure 23, HH always performed better compared to the BRC approach, regardless of the configuration.

This performance gain can vary significantly depending on the different scenarios tested (Figure 24 and Figure

25): from 0.77% for scenarios S2.1a and S2.1b up to 26.54% for S2.6a.

It also appears that in addition to being more efficient, HH is more resilient to the occurrence of breakdowns than BRCs, as shown in Figure 26.

2. *Impact of arrival rates:* as the results show, the performances of the HH are strongly correlated to the product arrival rate. However, as shown in Figure 27, ITV is particularly efficient with cases of heavily used

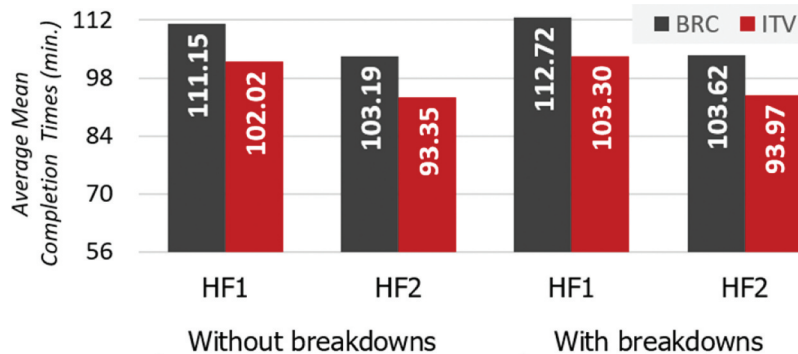


Figure 26. Impact of breakdowns on mean completion times.

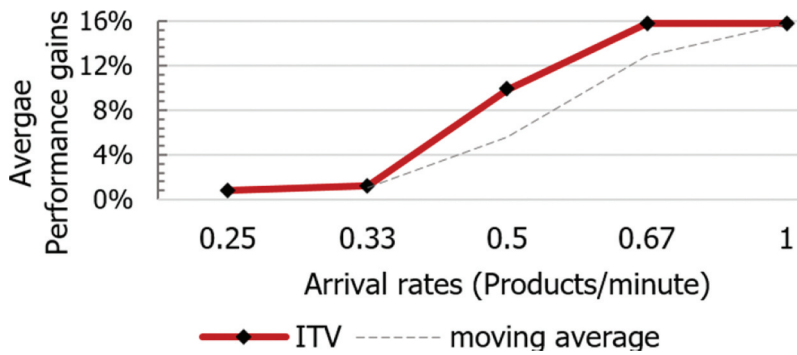


Figure 27. Average performance gains compared to arrival rates.

systems. Indeed, high arrival rates lead to SP accumulations within the system and the occurrence of bottlenecks. The ability of SP to combine several rules in the decisional strategies enables a more efficient flow of products compared to BRC. By taking into account the DC, the SP can select and apply the most relevant rule, notably taking into account the number of

products in the cell and the resource queue sizes. This contrasts with the behavior exhibited by the SP with BRCs, which highlights the fact that the perception of the DC induces a significant improvement in performance.

3. Impact of the precision of the decisional contexts: the proposed HH is based on a stochastic search and

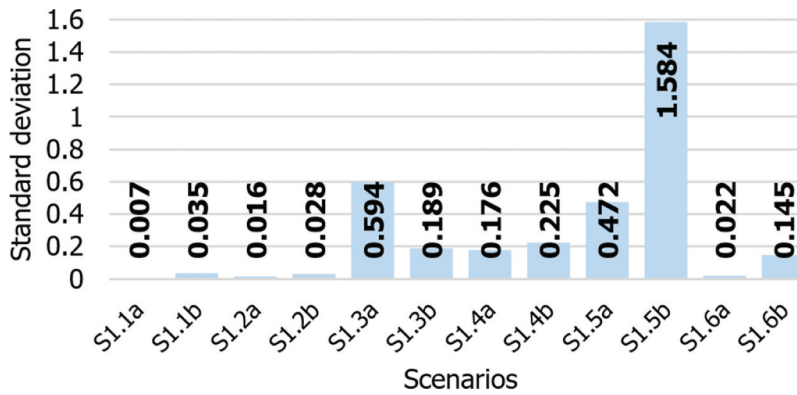


Figure 28. Standard deviations of the HH across HF1.

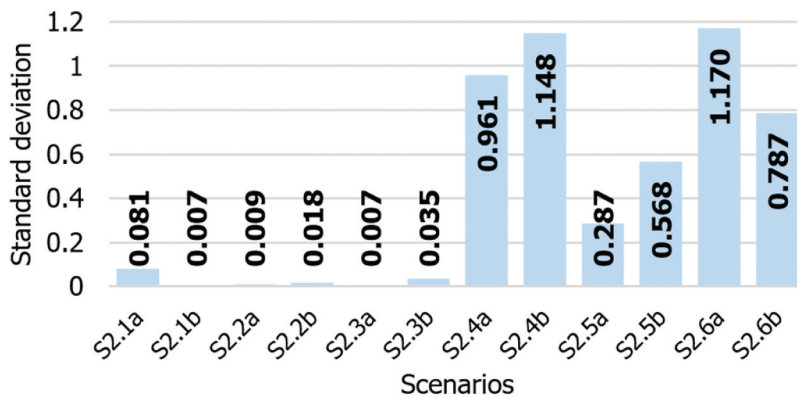


Figure 29. Standard deviations of the HH across HF2.

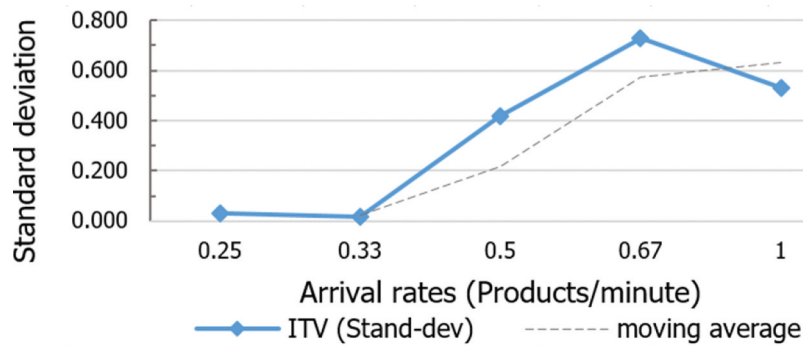


Figure 30. Average standard deviations versus arrival rates.

so it is important to consider the standard deviations of different executions. As exhibited in Figure 28 and Figure 29, variations were observed in the standard deviations of the HH. It also seems to be correlated with the arrival rate (Figure 30). In other words, even if the HH ends up converging towards efficient solutions, these results reveal a greater ‘difficulty of convergence’ over the 30 independent runs.

These standard deviations draw attention to the complexity of the decisional strategies implemented and highlight the possible limits of this type of approach when taking into account an ever-increasing number of decisional contexts. This is reminiscent of the problem of overfitting induced by an excessive inclusion of terms in the models that prove to be more complex than necessary (Hawkins 2004). The decisional strategies must remain well-balanced by allowing the behavior of the SP to be defined in detail while avoiding the combinatorial explosion of the search space.

From these experiments, to be pursued and intensified, the proposed PDCS integrating a HH was able to face perturbations reactively while ensuring the good global performance of the production system. However, the HH may come up against certain limits as the complexity of the decisional strategy, which can lead to convergence issues.

## 7. Conclusion

In this paper, the dynamic scheduling of manufacturing systems was addressed. A solution integrating a Product-Driven Control Architecture with a hyper-heuristic was proposed. A generic model of a decisional strategy using intervals of

discriminating variables allows Smart Products to characterize different decisional contexts and thus switch efficiently from one scheduling rule to another. An implementation was proposed for the dynamic scheduling of multi-stage manufacturing systems with multiple constraints (precedence, interoperability, setup times, etc.).

To validate the proposed approach, a case study inspired by the pharmaceutical industry and based on two different hybrid flexible flow shop configurations was tested using this implementation. A total of 24 problem instances were generated with different dynamic product arrivals and resource breakdowns. The comparative study with 168 combinations of classical scheduling rules from the literature highlighted the superiority of the proposed ITV strategy to minimize the Mean Completion Time. Furthermore, the proposed approach enhanced both the global performance and reactivity of the manufacturing control system. However, one limitation of our contribution is the complexity of the decisional strategies implemented. Indeed, the analysis of the standard deviations highlights that although intervals allow the products to efficiently switch from one rule to another, they also increase the search space to be explored by the hyper-heuristic.

These first encouraging results open up interesting prospects. Further work must also be carried out to develop more effective decisional contexts by exploring other discriminating variables. In addition, a future implementation on the S-mart/AIP-PRIMECA cell (UPHF 2020) is under consideration. This one will allow making experiments on a real flexible manufacturing cell with robots, shuttle conveyors, etc. Some dynamic

events such as maintenance tasks and some constraints such as transportation times must be addressed in the future as well.

## Note

1. <https://doi.org/10.13140/RG.2.2.30326.11848>.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## ORCID

Wassim Bouazza  <http://orcid.org/0000-0001-6290-5591>

Damien Trentesaux  <http://orcid.org/0000-0003-2489-6203>

## References

- Akkiraju, R., P. Keskinocak, S. Murthy, and F. Wu. 2001. "An Agent-based Approach for Scheduling Multiple Machines." *Applied Intelligence* 14 (2): 135–144. doi:10.1023/A:1008363208898.
- Anuradha, V.P. and Sumathi, D., 2015. "A survey on resource allocation strategies in cloud computing". In: 2014 International Conference on Information Communication and Embedded Systems, ICICES 2014. Chennai, India: IEEE, 1–7.
- Aydilek, H., and A. Allahverdi. 2012. "Heuristics for No-wait Flowshops with Makespan Subject to Mean Completion Time." *Applied Mathematics and Computation* 219 (1): 351–359. doi:10.1016/j.amc.2012.06.024.
- Blunck, H., D. Armbruster, and J. Bendul. 2018. "Setting Production Capacities for Production Agents Making Selfish Routing Decisions." *International Journal of Computer Integrated Manufacturing* 31 (7): 664–674. doi:10.1080/0951192X.2017.1379097.
- Borangiu, T., S. Raileanu, D. Trentesaux, T. Berger, and I. Iacob. 2014. "Distributed Manufacturing Control with Extended CNP Interaction of Intelligent Products." *Journal of Intelligent Manufacturing* 25 (5): 1065–1075. doi:10.1007/s10845-013-0740-3.
- Bouazza, W., D. Hamdadou, Y. Sallez, and D. Trentesaux. 2019. "Effective Dynamic Selection of Smart Products Scheduling Rules in FMS." *Manufacturing Letters* 20: 45–48. doi:10.1016/j.mfglet.2019.05.004.
- Bouazza, W., Sallez, Y., Aissani, N., and Beldjilali, B., 2015. A model for manufacturing scheduling optimization through learning intelligent products. In: *Studies in Computational Intelligence*. Cham, Switzerland: Springer International Publishing, 233–241.
- Branke, J., S. Nguyen, C. W. Pickardt, and M. Zhang. 2016. "Automated Design of Production Scheduling Heuristics: A Review." *IEEE Transactions on Evolutionary Computation* 20 (1): 110–124. doi:10.1109/TEVC.2015.2429314.
- Burke, E. K., M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu. 2013. "Hyper-heuristics: A Survey of the State of the Art." *Journal of the Operational Research Society* 64 (12): 1695–1724. doi:10.1057/jors.2013.71.
- Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J.R., 2010. A Classification of Hyper-heuristic Approaches. In: M. Gendreau and J.-Y. Potvin, eds. *Handbook of metaheuristics*. Boston, MA: Springer US, 449–468.
- Burke, E.K., Hyde, M.R., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J.R., 2019. A classification of hyper-heuristic approaches: Revisited. In: M.G. J.-Y. Potvin, ed. *International Series in Operations Research and Management Science*. New York, NY, USA: Springer, 453–477.
- Çakar, T., and I. Cil. 2004. "Artificial Neural Networks for Design of Manufacturing Systems and Selection of Priority Rules." *International Journal of Computer Integrated Manufacturing* 17 (3): 195–211. doi:10.1080/09511920310001607078.
- Carlucci, D., P. Renna, S. Materi, and G. Schiuma. 2020. "Intelligent Decision-making Model Based on Minority Game for Resource Allocation in Cloud Manufacturing." *Management Decision* 58 (11): 2305–2325. doi:10.1108/MD-09-2019-1303.
- Chen, G., P. Wang, B. Feng, Y. Li, and D. Liu. 2020. "The Framework Design of Smart Factory in Discrete Manufacturing Industry Based on Cyber-physical System." *International Journal of Computer Integrated Manufacturing* 33 (1): 79–101. doi:10.1080/0951192X.2019.1699254.
- Chern, C. C., and Y. L. Liu. 2003. "Family-based Scheduling Rules of a Sequence-dependent Wafer Fabrication System." *IEEE Transactions on Semiconductor Manufacturing* 16 (1): 15–24. doi:10.1109/TSM.2002.807742.
- Chiang, T.-C., and L.-C. Fu. 2009. "Using a Family of Critical Ratio-based Approaches to Minimize the Number of Tardy Jobs in the Job Shop with Sequence Dependent Setup Times." *European Journal of Operational Research* 196 (1): 78–92. doi:10.1016/j.ejor.2007.12.042.
- Choi, B. K., and N. K. You. 2006. "Dispatching Rules for Dynamic Scheduling of One-of-a-kind Production." *International Journal of Computer Integrated Manufacturing* 19 (4): 383–392. doi:10.1080/09511920500407541.
- Corning, P. A. 2002. "The Re-emergence of "Emergence": A Venerable Concept in Search of A Theory." *Complexity* 7 (6): 18–30. doi:10.1002/cplx.10043.
- Cowling, P., Kendall, G., and Soubeiga, E., 2001. "A hyperheuristic approach to scheduling a sales summit". In: E. Burke and W. Erben, eds. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Verlag, 176–190.
- Fattahi, P., and A. Fallahi. 2010. "Dynamic Scheduling in Flexible Job Shop Systems by considering Simultaneously Efficiency and Stability." *CIRP Journal of Manufacturing Science and Technology* 2 (2): 114–123. doi:10.1016/j.cirpj.2009.10.001.
- Framinan, J.M., Leisten, R., and Ruiz García, R., 2014. "Manufacturing Scheduling Systems". 1st ed. *Manufacturing*

- Scheduling Systems: An Integrated View on Models, Methods and Tools*. London: Springer London.
- Govett, J. 1965. "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility." *Management Science* 11 (8): 166–176.
- Hawkins, D. M. 2004. "The Problem of Overfitting." *Journal of Chemical Information and Computer Sciences* 44 (1): 1–12.
- Henderson, D., Jacobson, S.H., and Johnson, A.W., 2006. "The Theory and Practice of Simulated Annealing". In: F. Glover and G.A. Kochenberger, eds. *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 287–319.
- Herrera, C., Berraf, S.B., and Thomas, A., 2012. "Viable System Model Approach for Holonic Product Driven Manufacturing Systems". In: *Service Orientation in Holonic and Multi-Agent Manufacturing Control*. Berlin, Heidelberg: Springer, 169–181.
- Hershauer, J. C., and R. J. Ebert. 1975. "Search and Simulation Selection of a Job-Shop Sequencing Rule." *Management Science* 21 (7): 833–843.
- Jorapur, V. S., V. S. Puranik, A. S. Deshpande, and M. Sharma. 2016. "A Promising Initial Population Based Genetic Algorithm for Job Shop Scheduling Problem." *Journal of Software Engineering and Applications* 9 (5): 208–214.
- Kalinowski, K., Krenczyk, D., and Grabowik, C., 2013. "Predictive - Reactive Strategy for Real Time Scheduling of Manufacturing Systems". *Applied Mechanics and Materials* 307: 470–473.
- Keddis, N., G. Kainz, and A. Zoitl. 2015. "Product-driven Generation of Action Sequences for Adaptable Manufacturing Systems." *IFAC-PapersOnLine* 28 (3): 1502–1508.
- Kim, D. Y., J. W. Park, S. Baek, K. B. Park, H. R. Kim, J. I. Park, H. S. Kim, B. B. Kim, H. Y. Oh, K. Namgung, and W. Baek. 2020. "A Modular Factory Testbed for the Rapid Reconfiguration of Manufacturing Systems." *Journal of Intelligent Manufacturing* 31 (3): 661–680.
- Lei, J., and Z. Y. Yang. 2013. "Disturbance Management Design for a Holonic Multiagent Manufacturing System by Using Hybrid Approach." *Applied Intelligence* 38 (3): 267–278.
- Leitão, P. 2009. "Agent-based Distributed Manufacturing Control: A State-of-the-art Survey." *Engineering Applications of Artificial Intelligence* 22 (7): 979–991.
- Liou, C. D., and Y. C. Hsieh. 2015. "A Hybrid Algorithm for the Multi-stage Flow Shop Group Scheduling with Sequence-dependent Setup and Transportation Times." *International Journal of Production Economics* 170: 258–267.
- Mascia, F., M. López-Ibáñez, J. Dubois-Lacoste, and T. Stützle. 2013. "From Grammars to Parameters: Automatic Iterated Greedy Design for the Permutation Flow-shop Problem with Weighted Tardiness." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7997 (LNCS): 321–334.
- Morariu, O., Raileanu, S., Morariu, C., and Borangiu, T., 2014. "Multi-agent system for heterarchical product-driven manufacturing". In: *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*. IEEE, 1–6.
- Mousakhani, M. 2013. "Sequence-dependent Setup Time Flexible Job Shop Scheduling Problem to Minimise Total Tardiness." *International Journal of Production Research* 51 (12): 3476–3487.
- Nguyen, S., M. Zhang, M. Johnston, K. C. Tan, S. Member, M. Johnston, K. C. Tan, and S. Member. 2013. "A Computational Study of Representations in Genetic Programming to Evolve Dispatching Rules for the Job Shop Scheduling Problem." *IEEE Transactions on Evolutionary Computation* 17 (5): 621–639.
- Ochoa, G., Qu, R., and Burke, E.K., 2009. "Analyzing the landscape of a graph based hyper-heuristic for timetabling problems". In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO '09*. New York, New York, USA: ACM Press, 341–348.
- Ouelhadj, D., and S. Petrovic. 2009. "A Survey of Dynamic Scheduling in Manufacturing Systems." *Journal of Scheduling* 12 (4): 417–431.
- Pan, Q.-K., L. Gao, X.-Y. Li, and K.-Z. Gao. 2017. "Effective Metaheuristics for Scheduling a Hybrid Flowshop with Sequence-dependent Setup Times." *Applied Mathematics and Computation* 303: 89–112.
- Pannequin, R., G. Morel, and A. Thomas. 2009. "The Performance of Product-driven Manufacturing Control: An Emulation-based Benchmarking Study." *Computers in Industry* 60 (3): 195–203.
- Panwalkar, S. S., and W. Iskander. 1977. "A Survey of Scheduling Rules." *Operations Research* 25 (1): 45–61.
- Pérez-Rodríguez, R. and Hernández-Aguirre, A., 2018. "A hybrid estimation of distribution algorithm for flexible job-shop scheduling problems with process plan flexibility". *Applied Intelligence* 48 (10): 3707–3734.
- Pickardt, C. W., and J. Branke. 2012. "Setup-oriented Dispatching Rules - A Survey." *International Journal of Production Research* 50 (20): 5823–5842.
- Sallez, Y., T. Berger, D. Deneux, and D. Trentesaux. 2010. "The Lifecycle of Active and Intelligent Products: The Augmentation Concept." *International Journal of Computer Integrated Manufacturing* 23 (10): 905–924.
- Shahzad, A., and N. Mebarki. 2016. "Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation." *Computers* 5 (1): 3.
- Thomas, A., D. Trentesaux, and P. Valckenaers. 2012. "Intelligent Distributed Production Control." *Journal of Intelligent Manufacturing* 23 (6): 2507–2512.
- Trentesaux, D., C. Pach, A. Bekrar, Y. Sallez, T. Berger, T. Bonte, P. Leitão, and J. Barbosa. 2013. "Benchmarking Flexible Job-shop Scheduling and Control Systems." *Control Engineering Practice* 21 (9): 1204–1225.
- UPHF, 2020. "AIP-Priméca-Valenciennes [Online]." [Accessed 10 March 2021. Available from: <http://www.uphf.fr/aipnpdc/sites/fr.aipnpdc/files/pdf/equipement-plateforme-valenciennes.pdf>
- Vázquez Rodríguez, J.A., Petrovic, S., and Salhi, A., 2007. "A Combined Meta-Heuristic with Hyper-Heuristic Approach to the Scheduling of the Hybrid Flow Shop with Sequence Dependent Setup Times and Uniform Machines". In:



- Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications*. Paris, France, 506–513.
- 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.
- Windt, K., T. Becker, O. Jeken, and A. Gelessus. 2010. "A Classification Pattern for Autonomous Control Methods in Logistics." *Logistics Research* 2 (2): 109–120.
- Zambrano Rey, G., T. Bonte, V. Prabhu, and D. Trentesaux. 2014. "Reducing Myopic Behavior in FMS Control: A Semi-heterarchical Simulation-optimization Approach." *Simulation Modelling Practice and Theory* 46: 53–75.
- Zhang, H., and U. Roy. 2019. "A Semantics-based Dispatching Rule Selection Approach for Job Shop Scheduling." *Journal of Intelligent Manufacturing* 30 (7): 2759–2779.
- Zhang, Y., J. Wang, S. Liu, and C. Qian. 2017. "Game TheoryBased Real-Time Shop Floor Scheduling Strategy and Method for Cloud Manufacturing." *International Journal of Intelligent Systems* 32 (4): 437–463.