Computers & Industrial Engineering 190 (2024) 110006

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Long-sighted dispatching rules for manufacturing scheduling problem in Industry 4.0 hybrid approach

Emma Salatiello *, Silvestro Vespoli, Guido Guizzi, Andrea Grassi

Università degli Studi di Napoli Federico II, Dipartimento di Ingegneria Chimica, dei Materiali e della Produzione Industriale, Piazzale Tecchio 80, Napoli, Italy

ARTICLE INFO

Keywords:
Scheduling
Decentralized
Mass customization
Manufacturing
Industry4.0

ABSTRACT

The introduction of Cyber–Physical Systems, the implementation of machine learning, and the proposal of other technologies are some of the main features of Industry 4.0, which aims to change the way of production by introducing more flexibility and more autonomous decision-making by machines. The presence of centralized and decentralized systems and the continuous change in the market scenario has led to a transition of production systems from highly centralized or decentralized processes to a hybrid architecture that incorporated advantages of the production control approaches. The analysis of the production performances of recent semi-heterarchical architectures encourage to assess these architectures by introducing new scheduling approach that overcome the risk to take a myopic perspective: the long-sighted approach. To this aim, two new dispatching rules are proposed, both evaluated with different variability. This is significant because it allows the measurement of the performance of the production system, as well as the definition of the reactive and proactive capacity of semi-heterarchical architectures in an increasingly competitive market.

1. Introduction

The manufacturing industry is a crucial sector that transforms raw materials into finished products, driving economic growth and innovation.

In today's globalized and fast-paced business environment, organizations in this sector must continuously evolve and adapt to remain competitive. One of the latest technological advances in this field is Industry 4.0 (I4.0), which integrates digital technologies and a global perspective to create a more connected and automated manufacturing industry. This paradigm shift is expected to bring numerous benefits, including increased efficiency, cost reduction, and improved quality (Castelo-Branco et al., 2019; Popolo et al., 2021). However, the adoption of I4.0 technologies and practices requires significant investments and changes to existing production systems, which can be a daunting task for many organizations.

To overcome the challenges of the current industrial context, organizations must have the ability to allocate resources quickly and effectively, continually assessing their management strategies and production systems (Zimmermann et al., 2021). A key element in achieving these goals is the use of Manufacturing Planning and Control (MPC) systems. MPC systems play a central role in optimizing production processes by efficiently utilizing resources, maximizing service levels for customers, and improving production speed, flexibility, and adherence to delivery plans (Hazır et al., 2015).

In the literature, different MPC approaches have been investigated, and new ones are being developed with the emergence of I4.0 technologies (Bendul & Blunck, 2019). The two main approaches are centralized and decentralized systems. Centralized systems are known for their lack of flexibility to change, whereas decentralized systems offer greater flexibility, but also pose a risk of achieving a local optimum rather than a global one (Guizzi et al., 2017). As a result, researchers started to explore alternative hybrid MPC architecture structures, able to combine the advantages of both approaches (Brennan & Norrie, 2001; Vespoli et al., 2019). Several hybrid architectures have been proposed in literature, such as the Dynamic Hybrid Control Architecture (D-HCA) proposed by Jimenez et al., the Agent-based Semi-heterarchical control proposed by Sallez et al. and the Semi-heterarchical proposed by Grassi et al.

Specifically, the D-HCA consists of three functional layers: (i) the Global Decisional Entity (GDE), (ii) the Resource Decisional Entity (RDE), and (iii) the Local Decisional Entity (LDE). The GDE is responsible for global layer coordination, while the RDE and the LDE represent the operational layer in terms of the local production system (jobs) and resources (Jimenez et al., 2017). Similarly, the Agent-based semi-heterarchical control architecture proposed by Sallez et al. consists of three functional components: (i) the Enterprise Resource Planning (ERP), (ii) the Dynamic Allocation Processes (DAP), and (iii) the Dynamic Routing Processes (DRP) (Sallez et al., 2010). This architecture

* Corresponding author.
E-mail address: emma.salatiello@unina.it (E. Salatiello).

https://doi.org/10.1016/j.cie.2024.110006

Available online 24 February 2024

has a similar structure of the Semi-heterarchical presented by Grassi et al. also consisting of three functional levels: (i) the Knowledge-based Enterprise Resource Planning (KERP), (ii) the High Level Controller (HLC), and (iii) the Low Level Controller (LLC) (Grassi et al., 2020a). It is worth noting that both the architecture proposed by Sallez et al. and by Grassi et al. include a managerial control level at the top of the architecture (the ERP for the Sallez et al.'s architecture and the KERP for the Grassi et al. one), as well as lower control levels with clearly defined separate decision-making quotas. With regard to the last two architectures presented, both of them have a control level in direct contact with the production system, whose main functionality and responsibility is to schedule the entry of jobs. To achieve this objective, this level uses a dispatching rule that, by observing the production system and the job order queue to be produced, establishes which is the best job to admit into production.

Dispatching rules play a crucial role in manufacturing planning and control systems, determining the sequence in which jobs are processed and allocated to resources like machines (Sai Sandeep et al., 2022). Efficient dispatching is critical for on-time delivery, minimizing overdue and inventory costs, and adapting to dynamic and stochastic environments (Zheng et al., 2020). Different dispatching control methods, including pull rules and composite dispatching rules, have been proposed to optimize performance and enhance collaboration between equipment units. These rules employ simple heuristics like First In, First Out (FIFO), Shortest Processing Time (SPT), or more advanced ones. Additionally, researchers have explored Reinforcement Learning (RL) and fuzzy rule knowledge-based systems to improve dispatching in manufacturing systems (Attajer et al., 2022; Chen & Wu, 2008).

However, despite the advanced state of RL, no RL algorithm proposed in the literature exhibits generally adaptable performance. Each algorithm relies on specific factors such as the reward function, state definition, training, and the data available for training whose acquisition can be expensive in a real production scenario. On the other side, dispatching rules represent a dynamic approach that easily adapts and flexible in various production contexts and their changes.

To this extent, Panzer et al., in their work built on the architecture of Sallez et al., and Grassi et al., introduce a hyper-heuristics control model within a semi-heterarchical production system. Notably, a key limitation identified is the training of the network itself (Panzer et al., 2023). Researchers have also applied RL for the selection of the dispatching rule within the production system, rather than as a substitute for the dispatching rules themselves (Marchesano et al., 2021; Min & Yih, 2003; Panzer et al., 2023). Therefore, further study of these rules is valuable both as a scheduling element for MPC systems and as a supporting element for new RL approaches in selecting dispatching for job sequencing (Gavgani et al., 2017).

Several studies have explored the impact of dispatching rules on MPC. Marchesano et al. proposed a Performance-Based Decentralized Dispatching Rule (PDDR) for flow shop production lines, considering machine condition and performance. This rule proved to be more flexible and practical than traditional ones (Marchesano et al., 2021). Lee et al. devises new dispatching rules for improved production scheduling in a printed circuit board manufacturing system (Lee et al., 2003). Grassi et al. propose a throughput control algorithm for a production system managed by a semi-heterarchical architecture and propose a decentralized dispatching approach for production scheduling. The algorithm focuses on achieving a specific throughput target by considering Work-In-Process (WIP) as the primary control lever (Grassi et al., 2020b). Guizzi et al., Salatiello et al. instead propose a dynamic dispatching rule that aims to improve the performance of a production system compliant Industry 4.0 architectures (Guizzi et al., 2017; Salatiello et al., 2022).

However, these rules come with a limitation. To identify the next job for admission into production, they scrutinize one job at a time, determining which one optimally enhances the production system performance. In practice, individual job evaluations occur in isolation, overlooking synergies or potential trade-offs emerging when multiple jobs closely interact within the production system. This constrained approach aims to pick the most suitable job in a specific scenario, resulting in a short-sighted approach and potentially suboptimal sequencing, leading to a reduction in production system performance.

To overcome this limitation, the present work aims to propose two novel dispatching rules used in hybrid MPC architectures by looking for more jobs. In particular, the Closed Loop Dispatch Control and the Dynamic-NEH. These rules, in deciding which job to admit into the production system, evaluate not only individual jobs but also their potential combinations, dynamically choosing multiple jobs to admit into the production system at each iteration. As a result, they have dual objectives: (i) mitigating the risk of local optimization and (ii) enhancing the system's predictability. The first rule, the Couple Loop, aims to minimize the deviation of processing times, pursuing a well-balanced production system. Line balancing is indeed one of the crucial steps in this decision-making process (Boysen et al., 2022; Dolgui & Battaı, 2013). By ensuring a fair distribution of workload and optimizing resource utilization, this rule is expected to enhance overall efficiency and system performance. The second rule, the Dynamic-Neh, on the other hand, focuses on minimizing the makespan (i.e., the total time required to complete a set of jobs). The effectiveness of these rules will undergo comprehensive scrutiny across various scenarios, subjecting them to rigorous evaluation to assess their performance and test their adaptability to diverse conditions.

In conclusion, this study introduces innovative dispatching rules designed to address prevalent limitations identified in existing literature, particularly focusing on overcoming the issue of myopic decision-making processes in hybrid Manufacturing Process Control (MPC) systems. Specifically, the research contributes to the field by:

- Developing novel dispatching rules by considering the dynamic selection of job combinations, specifically targeting the mitigation of short-sightedness;
- Adopting a scenario-based approach that diverges from conventional steady-state analysis by considering a fixed and finite set of jobs. This approach facilitates a more nuanced understanding of batch processing and work shift performance through transient state data analysis;

The remainder of the paper is organized as follows: Section 2 provides an overview of the problem being considered; Section 3 introduces the proposed dispatching rules within a semi-heterarchical context; Section 4 describes the experimental methodology used to evaluate the proposed dispatching rule while Section 5 presents and discuss the results obtained from the considered scenarios. Finally Section 6 summarizes the main contribution of the paper and discusses potential future research directions.

2. Problem statement

Central to this study is the semi-heterarchical architecture, a hybrid MPC architecture proposed by Grassi et al., which offers a novel approach to dynamic production control (Grassi et al., 2020a) (Fig. 1).

The above-mentioned semi-heterarchical architecture by Grassi et al. comprises three distinct functional levels:

- The Knowledge-based Enterprise Resource Planning (KERP) at the managerial level, orchestrating production orders and delineating objectives.
- The High-Level Controller (HLC), tasked with optimizing throughput and cycle time across the production line.
- The Low-Level Control (LLC), the physical heartbeat of the system, overseeing real-time status, job allocation, and dynamic order re-sequencing.

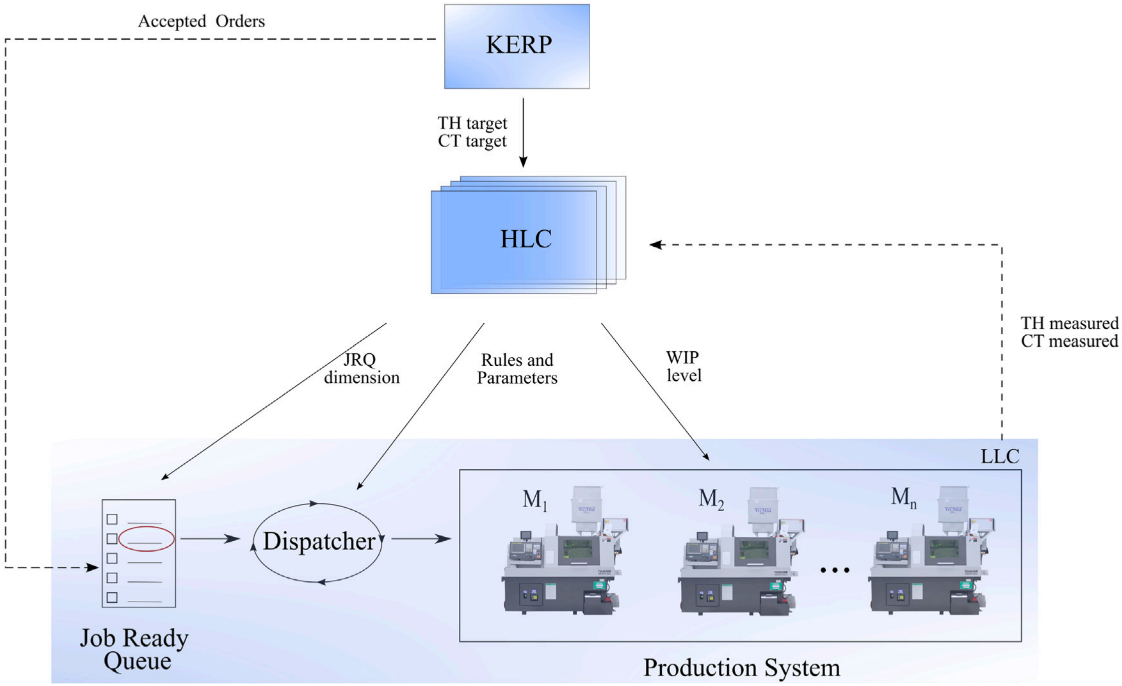

Fig. 1. The semi-heterarchical architecture inspired by Grassi et al. (2020a).

At the LLC level, the Job Ready Queue (JRQ), the "Dispatcher" and the production system are found. The JRQ operates as a controlled queue, holding orders accepted by managerial levels but not yet released for production. Regarding the production system, without losing generality, this study strategically adopts a flow-shop manufacturing system to align with existing literature, thereby facilitating a comparative analysis and a solid foundation for the assessment of novel dispatching rules. Among the various production control mechanisms developed in recent years by the literature, a CONstant-Work-In-Process (CONWIP) control system based on direct WIP control over the system has been chosen for the considered problem.

The Dispatcher, conversely, serves as the logical entity housing dispatching rules, responsible for selecting dynamically the next job from the JRQ to be admitted into the production system. In a more complex manufacturing system, there exist multiple HLC and LLC levels. However, for this paper, a scenario with a single HLC and a single LLC is considered.

In this context, Grassi et al. have proposed two dispatching rules at the LLC level, namely the Open Loop Control and Closed Loop Control dispatching rules, assessing their performance against a simple First-In-First-Out (FIFO) approach (Grassi et al., 2020c; Vespoli et al., 2019). Conversely, this study introduces two novel, long-sighted dispatching rules that operate within the same context, assessing them in a more comprehensive examination against these established rules under varied manufacturing scenarios. To this end, two primary control knobs are identified for the design of experimental scenarios: the distribution of the job processing times on machines and the type of feeding of the production system.

Specifically, the job processing times will be modelled using two different distributions:

- *Exponential distribution*, aligning with the Practical Worst Case scenario as identified by Hopp and Spearman in Hopp and Spearman (2011). This distribution reflects the variable nature of job processing times in a CONWIP system, offering insights into the system's behaviour and making it possible to compare against a well-known mathematical representation of the CONWIP system;
- *Uniform distribution* (refereed in the following as *Taillard*), as outlined by Taillard (1993), where job processing times at each station follow a uniform distribution $U[1..99]$. This scenario provides a contrasting benchmark, with the Nawaz–Enscore–Ham (NEH) algorithm serving as an additional performance yardstick for the proposed rules in this setting (Nawaz et al., 1983). To clarify, it is noteworthy that NEH algorithm is one of the best-performing rules for the considered problem in such a hypothesis; this rule, although not having a dynamic approach, will be used as an additional benchmark in this scenario to test the performance of the proposed rules (Framinan et al., 2001; Katragjini et al., 2013; Ruiz & Stützle, 2007).

Regarding the feeding of the production system, the study will explore two distinct scenarios:

- *Infinite Feeding*, inspired by the work of Grassi et al. in Grassi et al. (2021), to assess the steady-state performance of the system. This scenario is instrumental in understanding the steady-state performances of the production process under continuous operation.
- *Fixed Job Quantity*, focusing on a finite number of jobs, to evaluate the system's performance during the transient-state. This scenario, simulating the processing of a specific number of jobs (e.g., 100 jobs), is critical for assessing the system's responsiveness and adaptability to a defined workload, resembling real-world production shifts or batch processing.

Key performance indicators for the evaluation of the proposed rules include throughput (TH) — the rate at which jobs are produced —, Cycle Time (CT) — the average duration from the initiation to the completion of a job — and Work-in-Process (WIP) — representing the products in mid-production. In this context, the study aims to provide a comprehensive assessment of the dispatching rules within these varied manufacturing conditions, offering valuable insights into the scalability and flexibility of the semi-heterarchical architecture under both steady-state and transient operational states.

3. Proposed approaches

A hybrid Manufacturing Planning Control (MPC) system facilitates dynamic and adaptable operational decisions by assessing solutions that yield the most significant outcomes at each iteration. In this context, improving predictive capability would enhance performance compared to more narrowly focused rules, such as the Open Loop Dispatch Control and the Closed Loop Dispatch Control proposed by Grassi et al. in Grassi et al. (2020c), which rely on scheduling individual jobs.

For this reason, in this paper, two novel dispatching rules are proposed: Couple Loop Dispatch Control and Dynamic-NEH. The main aim is to overcome this limitation by adopting an approach that focuses on scheduling a larger number of jobs, aiming to provide a more 'long-sighted' perspective compared to Open Loop Dispatch Control and Closed Loop Dispatch Control (Grassi et al., 2020c). Both rules are implemented in a manufacturing context with a queue where jobs await entry into the production line, commonly known as a backlog. Since our work is related to the semi-heterarchical architecture where the production system is situated in the Lower Level Control, the backlog is represented by the logical entity Job Ready Queue (JRQ) (Fig. 1).

3.1. The couple loop dispatch control

Algorithm 1: The Couple Loop Dispatch Control

```
Data: The index of j jobs in the JRQ,the number of machines
      m the processing times of the jobs in the JRQ for each
      machine (t_{j,m}).
Result: The jobs scheduling with the Couple Loop Dispatch
        Control
/* Constructing a matrix [i] × [m], with i equals to
   the size of the JRQ                                  */
for Each row j(1 ≤ j ≤ i) do
    for Each k(1 ≤ k ≤ m) do
        Take the processing time of job j in the machine k, t_{j,k}
    end
    for Each column k do
        Evaluate the sum of processing time for each machine,
        T_k
    end
    for each element of the matrix, [i] × [m], t_{j,k} do
        Subtract T_k
    end
end
for Each row j do
    pick out D_i
end
for Each couple of element of vector D do
    Evaluate PM(J_i, J_y)
end
for Each PM(J_i, J_y) do
    Choose the job with min(PM(J_i, J_y) − PM(LC))^2
end
J_i, J_y are the jobs that best balance the workload along the
machines.
```

As previously said, the main aim of this dispatching rule is to be *long-sighted* by minimizing the processing time deviation and by balancing the production line. The Couple Loop Dispatch Control specifically identifies, within the Job Ready Queue (JRQ) of size i, the j jobs ($i \in (1,\dots,i)$) that better balance the production line workload compared to the last i jobs that entered the production line. This rule, in some aspects, extends the Open Loop Dispatch Control proposed by Grassi et al. (2020a). Although our discussion focuses on two jobs for practical reasons, it can be extended to involve more jobs.

To balance the workload in the production system, the aim is to select a couple of jobs in the JRQ that better balance the workload on the machine concerning the last couple of jobs entered into the production system. The algorithmic procedure is presented in Algorithm 1. The first step involves computing the mean Processing Time T_k for each machine k ($k \in (1,\dots,m)$) in the production system under consideration.

The notation is the following:

- i: the JRQ size
- j: the index of the jobs within the JRQ $j \in (1,\dots,i)$
- the mean Processing Time $T_k = \frac{(\sum_{j=1}^{i})t_{jk}}{i}$

The computation of T_k requires building a matrix of dimensions $[i] \times [m]$ in which each element represents the processing time of job j on machine k: t_{jk}. The second step involves calculating the deviation d_{jk}, which is the difference between t_{jk} and T_k.

$$d_{jk} = t_{jk} - T_k$$

A new $[i] \times [m]$ matrix is obtained, where each element is the corresponding d_{jk}. A vector D_j for the jth job is identified for each row of the matrix, with elements representing points in an m-dimensional Cartesian space.

$$D_j = (d_{j,1}, d_{j,2}, \dots, d_{j,m})$$

The choice of the couple of jobs is facilitated by calculating the midpoint vector $PM(J_i, J_y)$ for each couple of jobs within the JRQ.

$$PM(J_i, J_y) = [\frac{d_{i,1} - d_{y,1}}{2}, \dots, \frac{d_{i,m} - d_{y,m}}{2}]$$

At each iteration, the element of the midpoint vector $PM(LC)$ of the last couple of jobs entered into the system is subtracted from each element of the current midpoint vector $PM(J_i, J_y)$. The couple of jobs associated with the minimum value of this difference is then selected, and the chosen jobs are assigned a priority equal to 1.

The application example illustrated in Fig. 2 demonstrates the algorithm with a JRQ size of 3 and a production line with two machines. The algorithm analyses two jobs at each iteration. From the deviations, points in the Cartesian space (a, b and c in Fig. 2) for the three jobs are obtained. The midpoint vector PM is calculated for each, and $PM(LC)$ represents the midpoint vector of the last pair of jobs to enter production. It is recalled that the objective is to balance the workload between the machines with a long-sighted approach. In the provided example, the optimum point is $PM(*)$, and the algorithm selects the two jobs whose midpoint is the shortest distance from $PM(*)$. The distances are visualized in yellow, blue, and green, with the result that the pair a, c is identified as the one that best balances the workload.

The distances shown:

$[PM(a,b) - PM(*)]^2$ in yellow
$[PM(a,c) - PM(*)]^2$ in blue
$[PM(b,c) - PM(*)]^2$ in green

3.2. The Dynamic-NEH

The logic of the Dynamic-NEH algorithm can be described in three main steps, inherited from the NEH algorithm, with the ultimate goal of defining the best schedule that minimizes the makespan. The steps are

1. Adding up the processing times of each job for all machines.

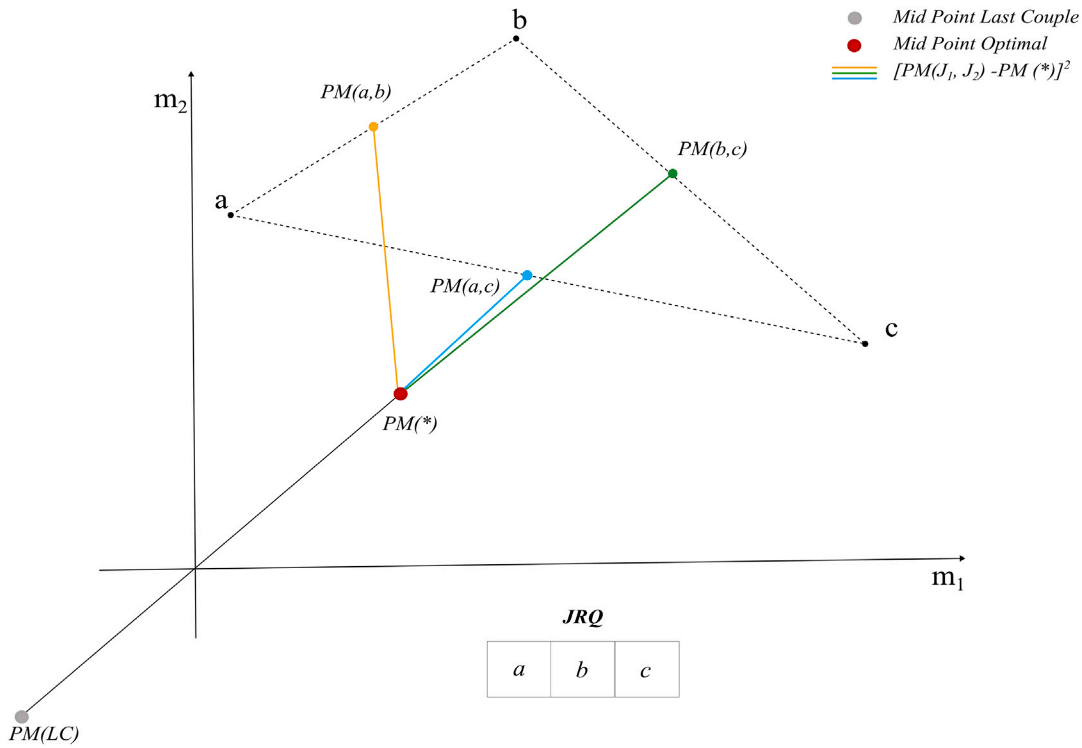

Fig. 2. The couple loop dispatch control.

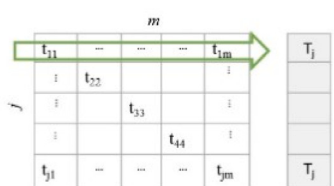

Fig. 3. Processing times sums matrix.

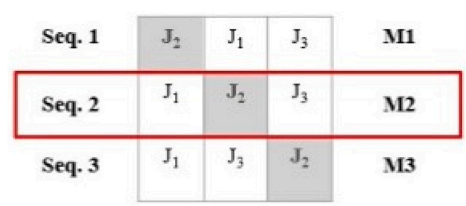

Fig. 4. Partial makespan.

2. Sort the jobs according to the sum of their decreasing processing times.
3. Choose jobs from the previously formed sorted sequence and arrange them in a way that minimizes makespan.

The algorithmic procedure is shown in Algorithm 2.

The difference between classic NEH and Dynamic-NEH is that the former schedules all jobs in the backlog at once, sending them to the production line in a predefined and deterministic sequence. In contrast, the proposed Dynamic-NEH optimizes the sequence each time a job leaves the backlog (the JRQ) and a new job enters the JRQ. This process involves evaluating the placement of multiple jobs in the JRQ with a dynamic and long-sighted approach

According to the steps, first, the job processing time $t_{j,m}$ of job j on each machine m is added up, as shown in Fig. 3.

$$T_j = \sum_{i=1}^{m} t_{ji}$$

with $(1 \leq \ i \leq) m$

with where T_j is the total processing time of the job j.

The calculation of T_j is possible by creating a matrix $[j] \times [m]$ with j equal to the JRQ size and m representing the size of the machines configured in the production system.

Hence, the sums T_j will be ordered successively in descending order in a vector $[1] \times [j]$.

By constructing the vector, the initial scheduling of jobs j is defined. The next step of the algorithm involves scheduling the jobs j to achieve sequencing that minimizes the makespan.

To obtain the optimal sequence, it is necessary to calculate the partial makespan, which involves determining the makespan of possible sequences each time a new job is added to the schedule set in the previous step.

To facilitate the calculation of the partial makespan, when constructing the matrix, it is important to place the new job in the diagonal of the matrix $[j] \times [j]$, and in the rest of the matrix elements, the jobs are inserted in the order in which they were previously scheduled, as shown in Fig. 4.

The choice of the partial schedule is determined by finding the minimum among M_1, M_2, M_3. For instance, if M_1 is the minimum, the sequence will be fixed in the order of *Sequence 2* in Fig. 4; J_1, J_2, J_3.

For the calculation of the partial makespan, a square matrix is constructed. The choice of the square matrix is dictated by the condition that each row corresponds to the various partial schedules on which the makespan is to be calculated. The matrix will have a dimension of $[i] \times [i]$, where $1 \leq i \leq j$, representing the number of jobs in the JRQ. It is constructed each time considering the order of the jobs already scheduled at that moment and adding the new job to the diagonal. This choice allows the matrix to be compiled quickly and neatly, and the makespan is calculated for each row j. The makespan calculation must take into account the processing time t_{ji}, where $1 \leq i \leq m$.

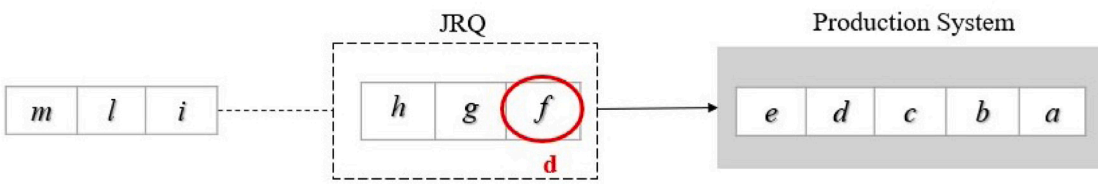

Fig. 5. The Dynamic-NEH: the job enter into the production line.

Algorithm 2: The Dynamic-NEH Algorithm

Data: The number of j jobs in the JRQ, the number of machines m, the processing times of the jobs in the JRQ for each machine ($t_{j,m}$).
Result: The jobs scheduling in the Dynamic-NEH Algorithm
for *Each row* j **do**
  for *Each* i *machine* **do**
    Insert the $t_{j,m}$
  end
  Evaluate the sum of the processing time $T_j = \sum_{i=1}^{m} t_{j,m}$
end
for *Each* $k(0 \le k \le j)$ **do**
  Select $max(T_k)$
end
Sort T_j in descending order
for *Each row* j **do**
  for *Each* $x(0 \le x \le j)$ **do**
    for *Each* $y(0 \le y \le j)$ **do**
      if $x = y$ **then**
        Insert new job
      else
        Insert the jobs according to the previously set schedule while maintaining the order
      end
    end
  end
  - Evaluate makespan
  - Select the Sequence j with the minimum makespan $S*$
end
$S*$ is the final scheduling of the Dynamic-NEH Algorithm.
/* when a job enters into the production line and a new job enters in the JRQ the algorithm is re-applied */
for *Each* $k(0 \le k \le j)$ **do**
  - Assign priorities
  - Check the Timeout
end

Once the minimum makespan is selected, the sequence of corresponding jobs is inserted into a vector of size i with $1 \le i \le m$. The vector increases in size each time a new job is inserted and a sequence is chosen. When the vector reaches a dimension of j, it is representative of the final Dynamic-NEH scheduling.

Figs. 5, 6 illustrate an application of Dynamic NEH at the ith iteration.

Initially, the JRQ contains the jobs h, g, and i. Upon entering the production system, job i is added to the JRQ, and the Dynamic-NEH algorithm selects the job schedule with the shortest makespan by calculating the various partial makespans associated with the schedules.

4. Experimental methodology

An in-depth analysis and evaluation of the proposed dispatching rules, namely Couple Loop Dispatch Control and Dynamic-NEH, has been conducted with a preliminary assessment and examination. To this end, a multi-method approach based on Discrete Event Simulation (DES) and Multi-Agent Systems (MAS) was used to develop the simulation model, using Anylogic 8.8.1 as simulation software (see Fig. 7).

In this, four types of agents were considered:

- The *Main Agent*: represents the development environment in which the HLC and JOBS population are generated;
- The *HLC Agent* is generated at the beginning of the simulation, according to the chosen workstation parameter which reproduces the second level of the Semi-Heterarchical structure
  - The *Machine Agent*: it resides in the HLC and reproduced the production line of the system
- The *Job Agent*: are generated by a static distribution; the jobs represent the items realized by the production system.

More specifically, the Production System (PS) starts production upon receiving an order, contingent upon the current number of orders being processed remaining below the permissible WIP limit. Subsequently, each instance a product (named as ‘Job’) exits the PS, a selection is made from the JRQ, and the chosen order is integrated into the PS. Concurrently, the ‘source’ block creates a new job whenever one is retrieved from the JRQ. Within this framework, machines are delineated as the entities tasked with the product’s processing. Jobs, in this context, refer to the items that are subjected to machining processes, which ultimately culminate in their transformation into finished products after the technological cycle. The model is distinguished by its high degree of parameterization, enabling the utilization of the tool to simulate scenarios with varying degrees of processing time variability by merely adjusting a select number of parameters. Additionally, it incorporates parameters that facilitate the simulation of production processes across a spectrum of job processing numbers and machine quantities. The agent-based modelling approach further enhances its versatility, rendering it eminently extendable for future research endeavours aimed at exploring a variety of operational dynamics.

Following the construction of the simulation model, it became important to ensure its emulation of a real system’s performance with consistency. Moreover, to guarantee reliable results, we conducted a validation phase against the previously mentioned PWC. During this evaluative stage, both the High-Level Control (HLC) control action and Dispatcher logic were temporarily suspended, thereby narrowing our focus exclusively to the throughput value (TH), expressed in jobs per hour, as stipulated by the PWC assumptions.

Following the evaluation of TH for various fixed WIP values, a t-Student test was employed to validate the simulator results. The simulation tool, through statistical analysis, demonstrated its capability to yield comparable mean values for both TH and CT as calculated by the PWC law proposed by Hopp and Spearman (2011) over a two-year simulation time run, with a confidence level of 95%. Importantly,

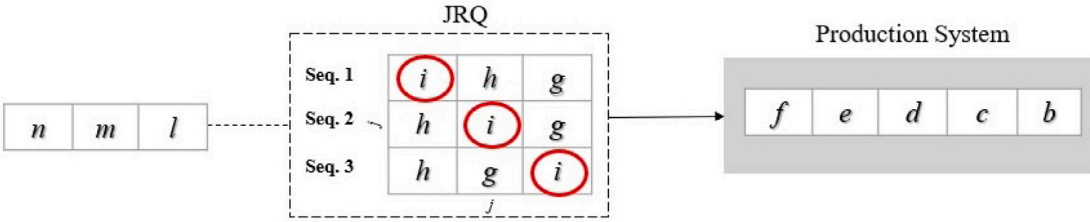

Fig. 6. The Dynamic-NEH: the job is scheduled.

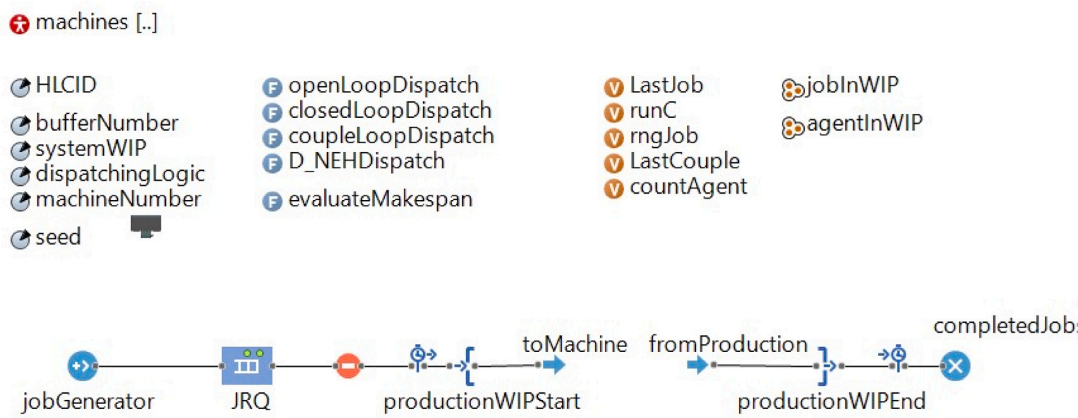

Fig. 7. Main agent of the simulation tool.

the Student's t-test, conducted to determine whether the means of the simulation data were significantly different from the theoretical one, failed to reject the null hypothesis at a 95% significance level. This confirms that the TH values obtained with the simulation tool align with the theoretical values, ensuring the adaptability to diverse conditions of our simulation results.

As outlined in the problem statement, the primary hypotheses focus on the processing time distributions for jobs on each machine. Specifically, the *Exponential* distribution with a mean value of 10 and a *Uniform* distribution $U[1 \dots 99]$ with a mean value of 50, following the *Taillard* approach, have been considered. The second major assumption pertains to two production system feeding scenarios: the first, *Infinite Feeding*, and the second, *Fixed Job Quantity*. For each of these combinations, and to assess the performance of the dispatching rule, different parameters have been varied, including the number of machines, WIP size, and JRQ size, as well as the different dispatching rules activated by the dispatcher.

The outcomes of these analyses will be interpreted within the context of a complete factorial plan, considering the parameters outlined for the *Infinite Feeding* in Table 1 and for the *Fixed Job Quantity* in Table 2.

- The *Infinite Feeding* main goal is to assess the production system performance under steady-state conditions. In each experiment, 40 replications have been carried out. The experimental campaigns for *Infinite Feeding* covered a time horizon of 4 years, with performance measurements logged every 4 h, representing each half-shift. (refer to Table 1).
- The *Fixed Job Quantity* scenario is for assessing the performance the dispatching rules and its production system during the transient state, which represents a specific number of jobs or a simulation equivalent to a single work shift. In each experiment, 200 replications have been carried out. All experimental campaigns within the *Fixed Job Quantity* scenario were executed until all generated jobs were processed. Performances have been measured every 4 h, corresponding to each half-shift (refer to Table 2).

It is crucial to note that in the *Fixed Job Quantity* scenario, when the number of jobs is set at 100 and the JRQ has a size of 100, the proposed Dynamic NEH, operates identically to the NEH one, especially noticeable with the makespan. This specific configuration allows for establishing a "deterministic" benchmark, to which the novel rules can be compared.

The analysis covers different performance indicators to comprehensively evaluate the effectiveness of dispatching rules. Specifically, the focus is on assessing throughput, as outlined in the 2, and the examination of coefficient variation of crossing time. For the *Fixed Job Quantity* scenario, additional analysis is applied to makespan. The choice of analysing the coefficient of variation is driven by the intention to assess the predictability of the proposed long-sighted dispatching rules in comparison to the short-sighted ones under examination. Crossing time is defined as the total waiting time of a job within the JRQ and the cycle time of the job within the production system and its coefficient of variation is computed as the ratio between the deviation and the mean (1).

$$c_v = \frac{\sigma}{\mu} \tag{1}$$

A dispatching rule that produces a lower coefficient of variation of crossing time generates a more predictive production system than one leading to a higher coefficient of variation. Pointing up the applied methodology, it is important to emphasize that each performance indicator will be analysed through its average value over a large number of experiment replications to ensure the adaptability to diverse conditions and validity of the obtained results.

5. Results and discussion

To thoroughly assess the performance of the production system across diverse simulated operating conditions, a series of experimental campaigns were conducted, according to the experimental methodology discussed above.

Table 1
Infinite feeding factorial plan.

Infinite feeding			
Processing time distribution	Exponential	Uniform (Taillard)	
Mean processing time	10 [min]	50 [min]	
JRQ size		5,10,15	
Machine number	5	10	15
System WIP	5 ÷ 15 in step 5	10 ÷ 50 in step 10	15 ÷ 60 in step 15
Dispatching rule	FIFO, Open, Closed, Couple, D-NEH		
Number of replication		40	

Table 2
Fixed job quantity factorial plan.

Fixed Job Quantity		
Processing time distribution	Exponential	Uniform (*Taillard*)
Mean processing time	10 [min]	50 [min]
Machine number	5,10,15	
JRQ size	10 ÷ 100 in step 10	
System WIP	10 ÷ 100 in step 10	
Dispatching rule	FIFO, Open, Closed, Couple, D-NEH	
Job quantity	100 [jobs]	
Number of replication	200	

5.1. The infinite feeding

5.1.1. The throughput analysis

To compare the results of the dispatching rules, the initial analysis focused on throughput performance. Several experimental campaigns were carried out based on the configurations outlined in Table 1. However, for practical presentation, Fig. 8 illustrates the throughput results horizontally, showcasing a production system with 15 machines and JRQ sizes of 5, 10, and 15. The WIP levels of 15, 30, 45, and 60 are arranged vertically. Simulations were replicated using both *Exponential* and *Uniform* distributions for generating processing times. It is important to note that the subsequent considerations are proportionally applicable to all the combinations listed in Table 1.

Fig. 8 illustrates that both proposed dispatching rules, the Couple and the Dynamic-NEH, outperform the other analysed dispatching rules across all given combinations. It highlights that the Couple rule, in a steady state, emerges as the most efficient rule, exhibiting a performance increase of approximately 5% with an Exponential Distribution and 2% with the Uniform Distribution compared to the other analysed dispatching rules. Meanwhile, the Dynamic-NEH shows a performance increase of 1% with an Exponential distribution.

These findings underscore how a long-sighted approach can lead to higher throughput performance, and the Couple Loop, by striving to balance jobs within the production system at each iteration, yields particularly promising results.

5.1.2. The crossing time analysis

The coefficient of variation for crossing time in a 15-machine manufacturing system is presented in Fig. 9. On the horizontal axis, WIP levels (15, 30, 45, 60) are depicted, while the values of JRQ size (5, 10, 15) are displayed upward in Fig. 9. The vertical axis represents the results in terms of c_v, with data generated using both Exponential and Uniform processing time distributions. In this scenario, Fig. 9 indicates that both proposed dispatching rules, particularly the Couple, yield the best outcomes. This trend becomes even more pronounced as the level of WIP increases for each JRQ size.

5.2. Fixed job quantity

Following the assessment of dispatching performance in the steady-state within the Infinite Feeding scenario, the analysis now shift to the Fixed Job Quantity scenario. This scenario enables the evaluation of performance during the transient state by simulating the execution of a small batch of jobs or a defined work shift. Specifically, the performance indicators analysed in the Infinite Feeding scenario will be replicated, with the addition of the makespan, which can be determined based on the defined quantity of finished jobs (100 in this work).

To accomplish this, several experimental campaigns were conducted following the parameters outlined in Table 2. For practical reasons, graphs related to selected combinations are presented; however, it is important to note that the considerations can be extended proportionally to all possible combinations of JRQ, WIP, dispatching, and processing time distribution.

5.2.1. The throughput analysis

The initial analysis of this scenario is presented in Fig. 10, illustrating the throughput performances of a 15-machine production system under various dispatching rules. The JRQ sizes are set at 10, 20, 60, and 100, and different WIP levels are configured to be 20, 60, and 100.

Once again, in this scenario, the proposed dispatching rules outperform the others, as shown in Fig. 10. Notably, in this transient state, the Dynamic-NEH exhibits the best performance, with a throughput improvement of around 25% over the short-sighted dispatching rules and approximately 20% over the Couple Loop in the Exponential Distribution. In the Uniform Distribution, it shows a performance increase of around 13%. Furthermore, a distinctive peak is observed when the JRQ size matches the lot size (i.e., 100 in this paper). In these scenarios, the Dynamic-NEH is comparable to the classic NEH algorithm, which requires complete knowledge of the jobs for scheduling.

As shown in Figs. 8 and 10, the proposed dispatching rules exhibit different trends depending on whether the system is in a steady state or transient state. This difference can be attributed to their ability to balance the production system. As previously mentioned, the Couple rule aims to minimize processing time deviations by balancing the production system at each iteration, leading to optimal performance values. However, it may struggle to achieve such optimization in the transient state, which requires a longer time interval to settle down, as illustrated in Fig. 11. This graph demonstrates that, except for the Dynamic-NEH, it is challenging to identify a consistent trend for various dispatching rules that clearly require a longer settling time. The configurations used in Fig. 10 are the same JRQ and WIP combinations.

On the other hand, the Dynamic-NEH, by minimizing the makespan at each iteration, can schedule jobs effectively during the transient state. This minimizes machine idle periods, thereby balancing the production system and optimizing throughput performance.

5.2.2. The crossing time analysis

The coefficient of variation for crossing time is the second index considered in both the Infinite Feeding and Fixed Job Quantity scenarios. In this case, the graph in Fig. 12 is presented, displaying results (on the vertical axis) from a production system consisting of 15 machines with a WIP level of 20, 60, and 100, and a JRQ size of 10, 20, 60, and 100. The different crossing time coefficients (c_v) for all possible combinations of WIP and JRQ defined earlier are represented on the horizontal axis. As can be seen, none of the dispatching rules exhibits a consistent trend; this is to be expected given the low quantity of jobs performed, which precludes noteworthy results in terms of predictability, while the dispatchings are in the transit state.

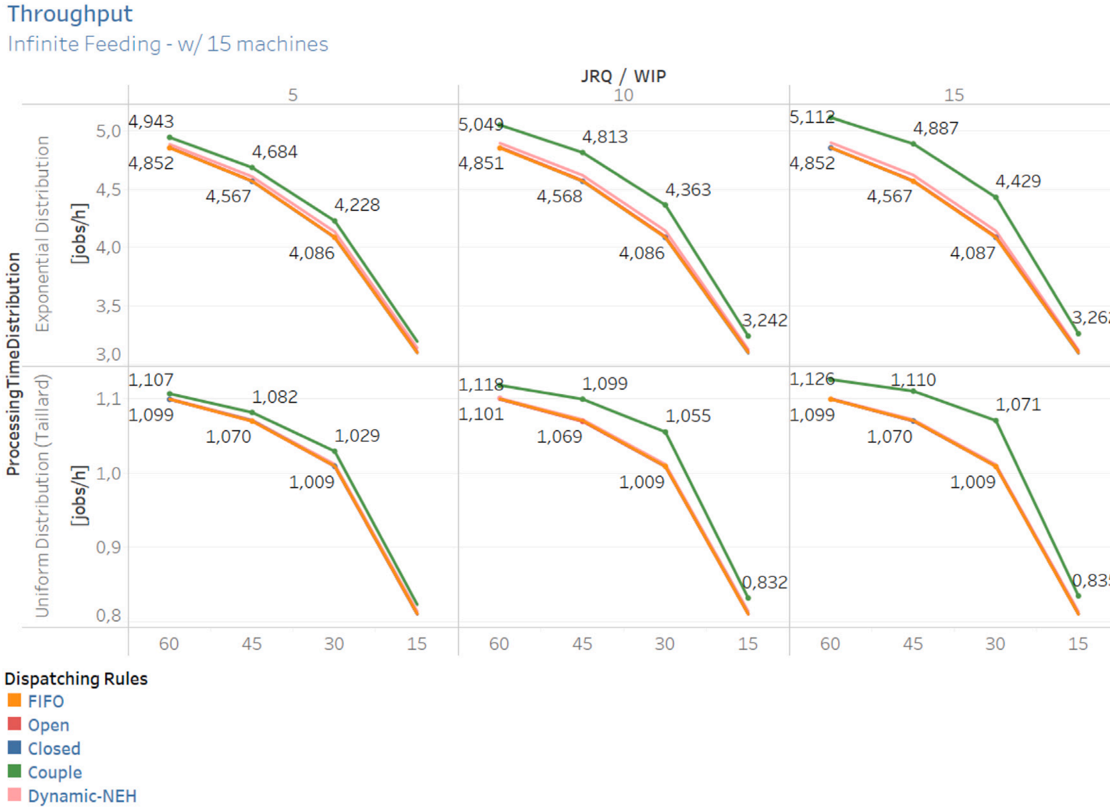

Fig. 8. The throughput in infinite feeding scenario.

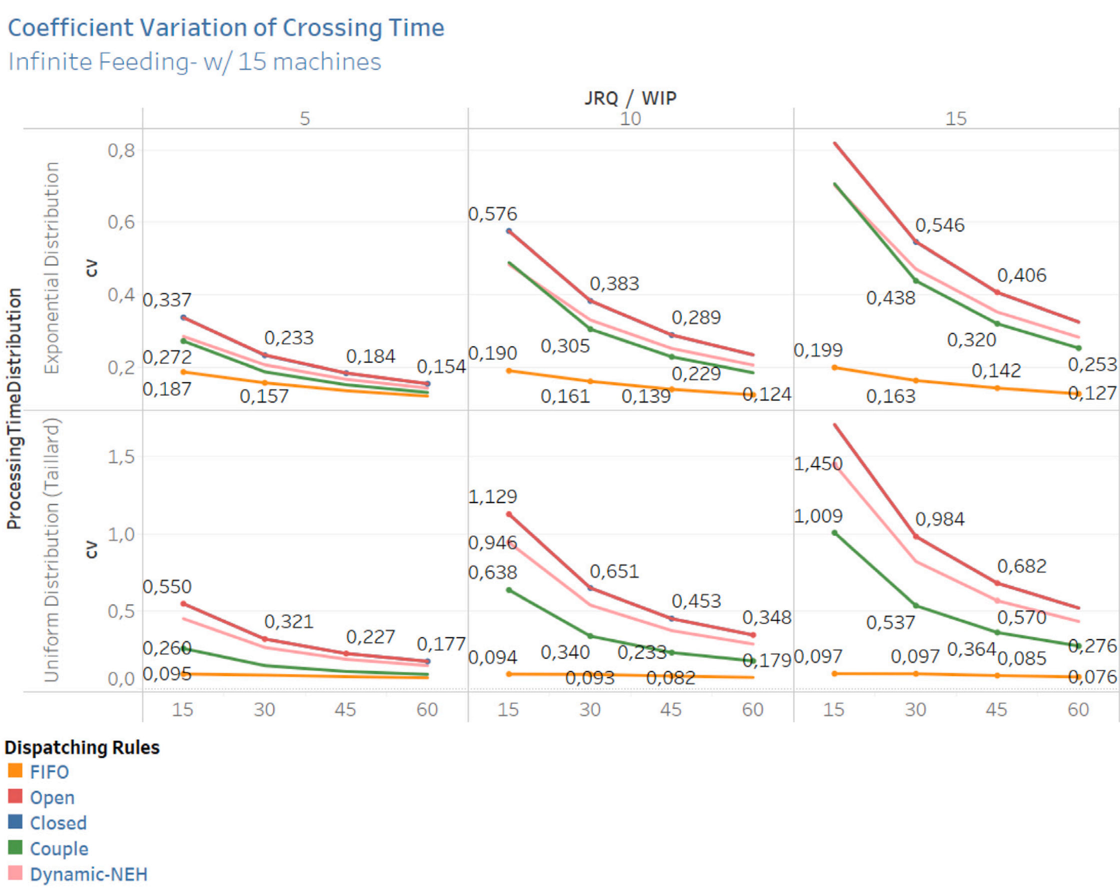

Fig. 9. The crossing time coefficient variation in infinite feeding scenario.

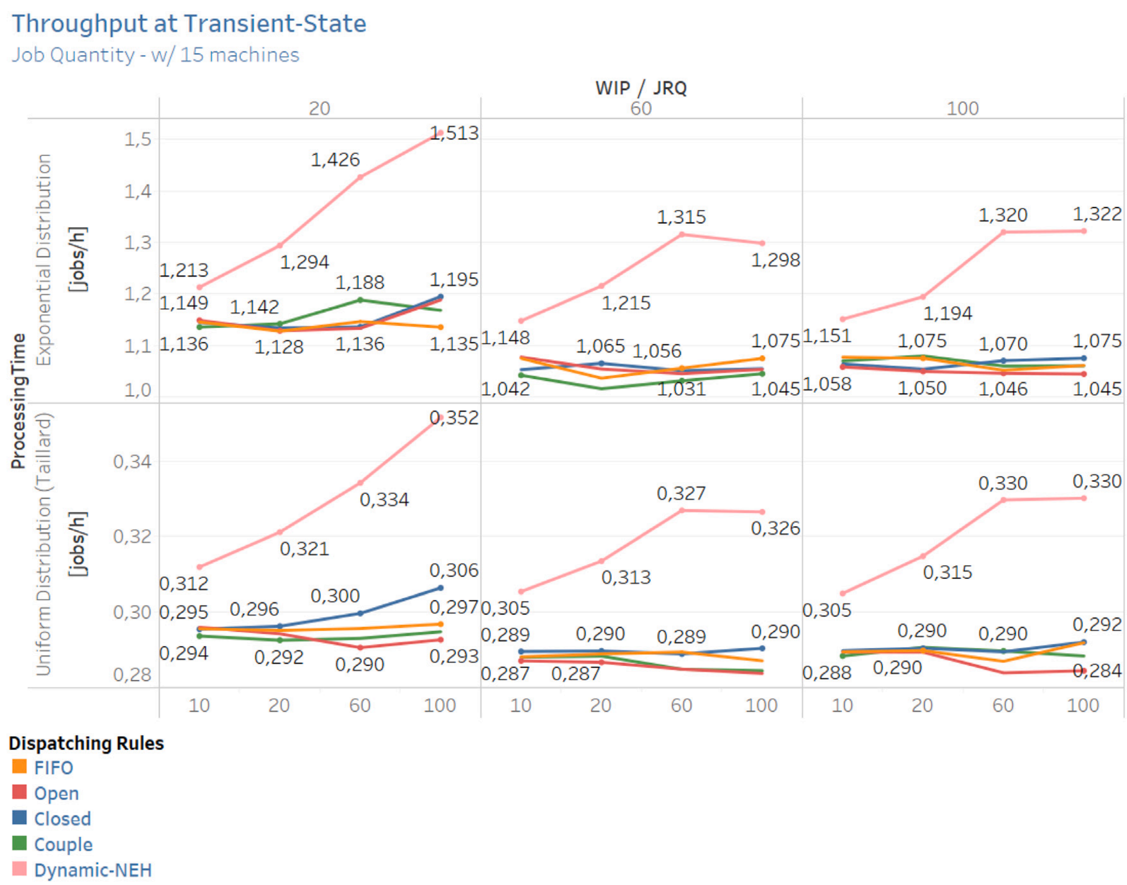

Fig. 10. The throughput in fixed job quantity scenario.

5.2.3. The makespan analysis

The makespan analysis for different dispatching rules has been performed by launching predefined job packages, each consisting of 100 jobs. With respect to various experimental campaigns related to the fixed amount of jobs outlined in Table 2, Fig. 13 presents the makespan values (on the vertical axis) for the production of 100 jobs in a 15-machine production system. The WIP and JRQ levels (on the horizontal axis) are set to 20, 60, 100, and 10, 20, 60, 100, respectively.

The dispatching rule that achieves significantly better results is Dynamic-NEH, it is indeed confirmed that the rule's dynamic operating logic effectively minimizes the makespan. This graph is particularly relevant as makes it possible a comparison of results obtained by different rules with a "deterministic" solution, which consistently performs better than the dynamic solutions mentioned in the other cases.

It is worth noting that with a JRQ size of 100, Dynamic-NEH is comparable to the NEH heuristic. In particular, the Table 3 shows the makespan values corresponding to the NEH algorithm for WIP of 20,60,100, each associated with its standard deviation. The standard deviations exhibit concentrated values relative to the mean, indicating stability in the results and an extended experimental campaign. Considering the case with WIP equals to 100, the "deterministic" solution resulting in 15% and 29% better, respectively, than the more dynamic solution at the same instance (i.e. 15 machines, WIP size 100, JRQ size 10) as showed in Fig. 13. These conditions, reflecting a scenario with a fixed order portfolio of 100 jobs and high rigidity, are, however, not always suitable for real production systems. Therefore, the adoption of a dynamic approach is crucial for adapting to current production contexts, even if it provides less performance value.

Table 3
Deviation standard of makespan.

JRQ size	WIP size	Makespan [min]	
		Exponential	Uniform
100	20	Mean 1602	6800
		Deviation 62,4	121,4
	60	Mean 1285	6181
		Deviation 49,9	109,9
	100	Mean 1293	6180
		Deviation 50,3	120,6

5.3. Synthesis of results

To summarize, the Couple Loop demonstrates a superior average throughput value in the steady state compared to the other analysed rules, whilst the Dynamic-NEH achieves the highest average throughput in the transient state. This can be attributed to the fact that, in the Infinite Feeding scenario, the Couple Loop reaches its peak functionality in the steady state. Conversely, this is not the case for the Fixed Job Quantity scenario, as the Couple cannot stabilize due to the limited intervention interval. In contrast, the Dynamic-NEH evaluates all jobs in the Job Release Queue (JRQ) from the initial iteration, yielding better results even during the transient state. It is consequently assumed that by repeating the experiment with a number of jobs greater

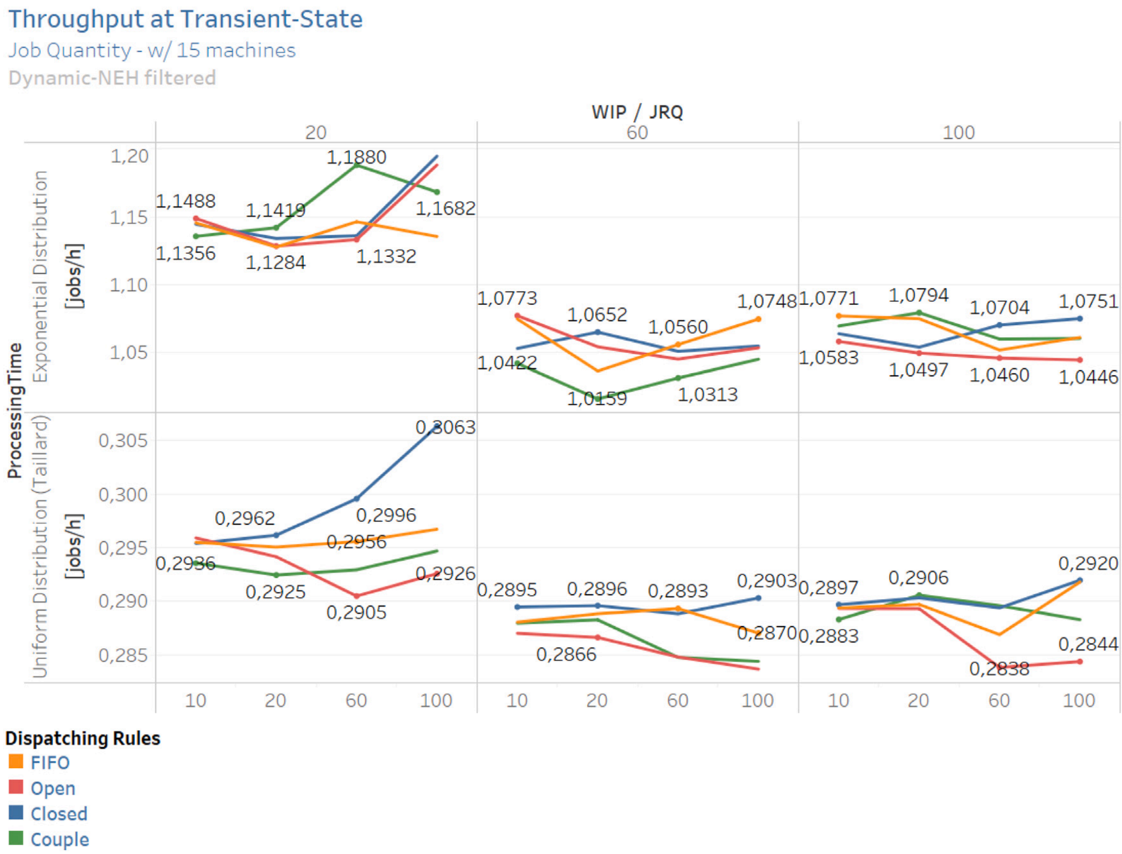

Fig. 11. The throughput in fixed job quantity scenario without the Dynamic-NEH.

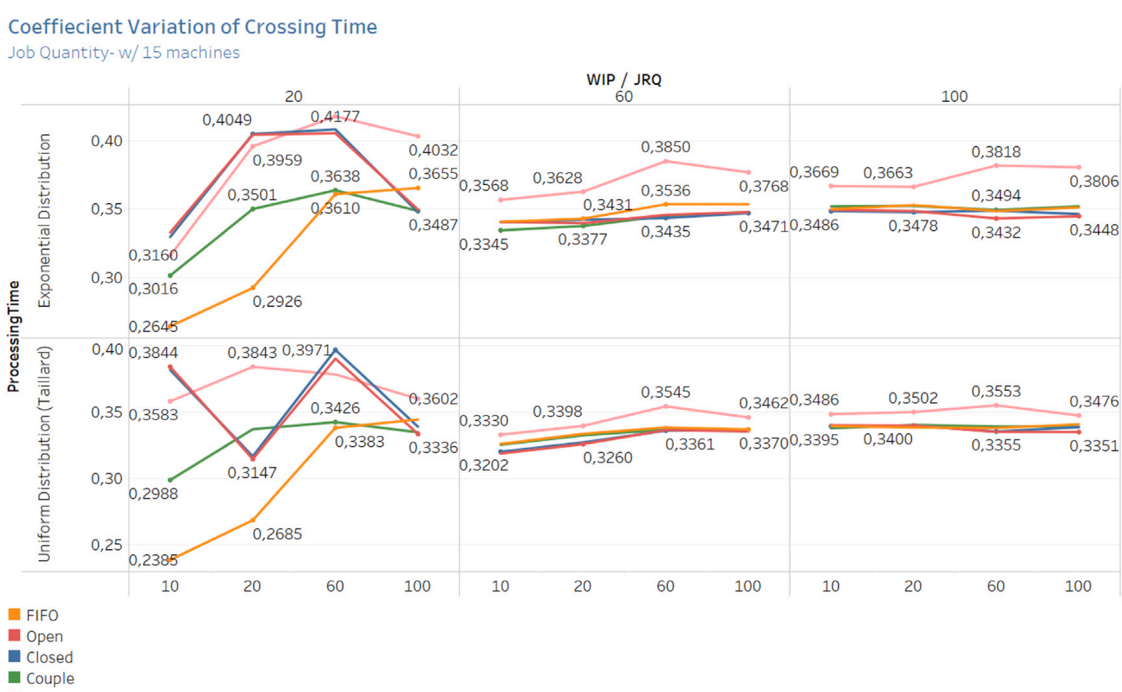

Fig. 12. The crossing time coefficient in fixed job quantity.

than 100 in the Fixed Job Quantity scenario, the Couple could obtain progressively better results.

Another vital performance indicator is the coefficient of variation of the crossing time. To better contextualize these findings, it is essential to recall that the crossing time constitutes the cumulative waiting time of jobs in the JRQ and the cycle time of jobs in the system. Relative to other rules, the FIFO consistently exhibits the lowest coefficient of variation. This rule prioritizes jobs based on their arrival

Fig. 13. The makespan.

order, enhancing predictability; owing to its simplicity, the impact of job waiting time in the JRQ is minimal. However, this variability is transferred to the production system, causing FIFO to demonstrate lower performance metrics (i.e., throughput) than other rules. Conversely, with the other dynamic control rules analysed (i.e., Open, Closed, Couple, and Dynamic-NEH), variability is shifted to the JRQ, wherein the next job for production is iteratively determined. This approach offers better throughput performance but a higher coefficient of variation of the crossing time compared to FIFO. Notably, the Couple Loop and Dynamic-NEH exhibit lower coefficient values, thereby establishing a more predictable production system against the classical FIFO approach, whilst concurrently achieving superior throughput performance compared to the best rules available in the literature.

Regarding the makespan in the Fixed Job Quantity scenario, a progressive improvement in the performance of the Dynamic-NEH is observed as the size of the Job Release Queue (JRQ) increases. Notably, in the limit case where both the number of jobs generated and the JRQ size are equal to 100, the Dynamic-NEH operates identically to the classical NEH rule, which represents the deterministic benchmark in this context. Additionally, when the JRQ size is less than 100, the performance of the Dynamic-NEH, though inferior to the deterministic NEH case, does not significantly deviate. This relative closeness in performance is noteworthy, suggesting that the Dynamic-NEH approach remains a compelling alternative to the classical NEH, particularly considering its benefits in terms of scheduling flexibility and computational efficiency. The enhanced flexibility allows for greater responsiveness to the arrival of new orders, while the reduced computational demand makes the approach more feasible and practical for extensive application. This balance between performance and operational agility renders the Dynamic-NEH an attractive option, even in scenarios where the deterministic NEH could theoretically provide superior results.

6. Conclusions

In conclusion, this paper has addressed the critical need for developing skills and capabilities to effectively allocate production resources in an increasingly dynamic manufacturing environment. This work centred on a hybrid semi-heterarchical architecture, tailor-made for decentralized production control. We have evaluated current dispatching rules for a Flow-Shop Production System with CONWIP logic, drawing upon the technological innovations spurred by the Fourth Industrial Revolution. Consequently, we proposed two new dispatching rules – the Couple Loop Dispatch Control and Dynamic NEH – which utilize dynamically required system information to define a production schedule, optimizing overall system performance.

Our analysis has been conducted on two levels. Initially, we evaluated the architecture's steady-state performance using the available dispatching rule from the literature in an extended test scenario. Subsequently, we assessed the transient-state performance of both proposed and known dispatching rules in a new scenario, derived from the assumptions of the Taillard dataset. For this purpose, we employed AnyLogic, a highly parametrizable multi-agent simulation tool, which enabled us to conduct various experimental campaigns.

Examining the performance of the proposed rules and the most advanced in the literature within the production system, it has emerged that in the steady-state, dynamic environment where jobs dynamically arise in the production system, the Couple Loop, in particular, outperforms traditional scheduling methods such as FIFO and rule-based control methods like Open Loop and Closed. Both new rules, Couple Loop and Dynamic-NEH, have consistently outperformed existing ones, aligning with their goal of optimizing performance and balancing the production system. On the other hand, in the transient state, Dynamic-NEH has demonstrated a more efficient evaluation of performances, in line with its objective of minimizing makespan at each iteration. This behaviour increases until the scheduling of static problems is represented when the JRQ and job quantity have the same size (100 in this study), where Dynamic-NEH becomes comparable with the classic NEH algorithm.

Examining the performance of different rules within the production system, it emerges that in scenario where jobs dynamically arrive in the production system, the Couple Loop outperforms traditional scheduling methods such as the FIFO and rule-based control methods available in the literature. On the other hand, in the transient state, Dynamic-NEH has demonstrated a more efficient evaluation of performances, in line with its objective of minimizing makespan at each iteration. This behaviour increases until the scheduling of static problems is represented when the JRQ and job quantity have the same size (100 in this study), where Dynamic-NEH becomes comparable with the classic NEH algorithm.

For future research, it would be valuable to delve into performance evaluations while incorporating disruption events, including aspects such as machine failure, within the context of Flow-Shop configuration. Additionally, there is room for further exploration into the development of a dispatching rule featuring a long-term strategy, specifically in a Hybrid Flow-Shop and Job-Shop configuration. In conclusion, this paper has made noteworthy progress in augmenting the efficiency of manufacturing systems. As result, based on the outcomes of the proposed rules, consideration is being given to implementing a system with multiple Low-Level Controllers (LLCs) operating concurrently, coupled with cooperative interactions among multiple KERP entities.

CRediT authorship contribution statement

Emma Salatiello: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Silvestro Vespoli:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Guido Guizzi:** Supervision, Methodology, Conceptualization. **Andrea Grassi:** Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Generative AI tool in order to proofread the manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Attajer, Ali, Darmoul, Saber, Chaabane, Sondes, Sallez, Yves, & Riane, Fouad (2022). An analytic hierarchy process augmented with expert rules for product driven control in cyber-physical manufacturing systems. *Computers in Industry*, *143*, Article 103742.

Bendul, Julia C., & Blunck, Henning (2019). Computers in Industry The design space of production planning and control for industry 4 . 0. *Computers in Industry*, [ISSN: 0166-3615] *105*, 260–272. http://dx.doi.org/10.1016/j.compind.2018.10.010.

Boysen, Nils, Schulze, Philipp, & Scholl, Armin (2022). Assembly line balancing : What happened in the last fifteen years ? *European Journal of Operational Research*, [ISSN: 0377-2217] *301*(3), 797–814. http://dx.doi.org/10.1016/j.ejor.2021.11.043.

Brennan, Robert W., & Norrie, Douglas H. (2001). Evaluating the performance of reactive control architectures for manufacturing production control. *Computers in Industry*, *46*(3), 235–245.

Castelo-Branco, Isabel, Cruz-Jesus, Frederico, & Oliveira, Tiago (2019). Assessing industry 4.0 readiness in manufacturing: Evidence for the European union. *Computers in Industry*, *107*, 22–32.

Chen, Juin-Han, & Wu, Yi-Sheng (2008). Dispatching rule in a fuzzy rule knowledge-based system for automated manufacturing. *International Journal of the Computer, the Internet and Management*, *16*(1), 53–60.

Dolgui, Alexandre, & Battaı, Olga (2013). Int . J . Production Economics A taxonomy of line balancing problems and their solution approaches. *International Journal of the Production Economics Journal*, *142*, 259–277. http://dx.doi.org/10.1016/j.ijpe.2012.10.020.

Framinan, Jose M., González, Pedro L., & Ruiz-Usano, Rafael (2001). Sequencing CONWIP flow-shops: Analysis and heuristics. *International Journal of Production Research*, [ISSN: 00207543] *39*(12), 2735–2749. http://dx.doi.org/10.1080/00207540110048972.

Gavgani, Saba Sadeghi, Zohrehbandian, Majid, & Sadeghi, Nima (2017). Selecting dispatching rule in manufacturing systems via DEA cross efficiency. *International Journal of Information and Decision Sciences*, *9*(2), 168–178.

Grassi, Andrea, Guizzi, Guido, Santillo, Liberatina Carmela, & Vespoli, Silvestro (2020a). The manufacturing planning and control system: A journey towards the new perspectives in industry 4.0 architectures. *International Series of the Operational Research and Management Science*, [ISSN: 22147934] *289*, 193–216.

Grassi, Andrea, Guizzi, Guido, Santillo, Liberatina Carmela, & Vespoli, Silvestro (2020b). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacture and Letters*, [ISSN: 22138463] *24*, 43–46. http://dx.doi.org/10.1016/j.mfglet.2020.03.007.

Grassi, Andrea, Guizzi, Guido, Santillo, Liberatina Carmela, & Vespoli, Silvestro (2020c). Assessing the performances of a novel decentralised scheduling approach in Industry 4.0 and cloud manufacturing contexts. *International Journal of Production Research*, [ISSN: 1366588X] 1–20. http://dx.doi.org/10.1080/00207543.2020.1799105.

Grassi, Andrea, Guizzi, Guido, Santillo, Liberatina Carmela, & Vespoli, Silvestro (2021). Assessing the performances of a novel decentralised scheduling approach in Industry 4.0 and cloud manufacturing contexts. *International Journal of Production Research*, *59*(20), 6034–6053.

Guizzi, Guido, Vespoli, Silvestro, & Santini, Stefania (2017). On the architecture scheduling problem of Industry 4.0. *CEUR Workshop Proceedings*, [ISSN: 16130073] *2010*, 94–100.

Hazır, Öncü, Delorme, Xavier, & Dolgui, Alexandre (2015). Annual Reviews in Control A review of cost and profit oriented line design and balancing problems and solution approaches. *Annual Reviews in Control Journal*, *40*, 14–24. http://dx.doi.org/10.1016/j.arcontrol.2015.09.001.

Hopp, Wallace J., & Spearman, Mark L. (2011). *Factory physics*. Waveland Press.

Jimenez, Jose Fernando, Bekrar, Abdelghani, Zambrano-Rey, Gabriel, Trentesaux, Damien, & Leitão, Paulo (2017). Pollux: a dynamic hybrid control architecture for flexible job shop systems. *International Journal of Production Research*, [ISSN: 1366588X] *55*(15), 4229–4247. http://dx.doi.org/10.1080/00207543.2016.1218087.

Katragjini, Ketrina, Vallada, Eva, & Ruiz, Rubén (2013). Flow shop rescheduling under different types of disruption. *International Journal of Production Research*, [ISSN: 00207543] *51*(3), 780–797. http://dx.doi.org/10.1080/00207543.2012.666856.

Lee, Geun-Cheol, Kim, Yeong-Dae, Kim, Jae-Gon, & Choi, Seong-Hoon (2003). A dispatching rule-based approach to production scheduling in a printed circuit board manufacturing system. *Journal of the Operational Research Society*, *54*(10), 1038–1049.

Marchesano, Maria Grazia, Guizzi, Guido, Santillo, Liberatina Carmela, & Vespoli, Silvestro (2021). Dynamic scheduling in a flow shop using deep reinforcement learning. In *IFIP international conference on advances in production management systems* (pp. 152–160). Springer.

Min, Hyeung-Sik, & Yih, Yuehwern (2003). Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. *International Journal of Production Research*, *41*(16), 3921–3941.

Nawaz, Muhammad, Enscore, E. Emory, Jr., & Ham, Inyong (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95.

Panzer, Marcel, Bender, Benedict, & Gronau, Norbert (2023). A deep reinforcement learning based hyper-heuristic for modular production control. *International Journal of Production Research*, 1–22.

Popolo, Valentina, Gallo, Mose, Grassi, Andrea, & Marchesano, Maria Grazia (2021). Exploiting the Full Potential of I4.0 Technologies for Products EOL Recovery Process. In *IFIP adv. inf. commun. technol.*: *Vol. 632 IFIP*, (pp. 307–316). Springer International Publishing, ISBN: 9783030859053, URL http://dx.doi.org/10.1007/978-3-030-85906-0{_}35.

Ruiz, Rubén, & Stützle, Thomas (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, [ISSN: 03772217] *177*(3), 2033–2049. http://dx.doi.org/10.1016/j.ejor.2005.12.009.

Sai Sandeep, K, Nageswara Rao, M, Rakesh, K, Phanindra Kshatra, D, & Ravi Teja, KMV (2022). Operational control decisions through random rule in flexible manufacturing system. In *Technology innovation in mechanical engineering: select proceedings of TIME 2021* (pp. 941–950). Springer.

Salatiello, Emma, Guizzi, Guido, Marchesano, Maria Grazia, & Santillo, Liberatina Carmela (2022). Assessment of performance in Industry 4.0 enabled Job-Shop with a due-date based dispatching rule. *IFAC-PapersOnLine*, *55*(10), 2635–2640.

Sallez, Y., Berger, T., Raileanu, S., Chaabane, S., & Trentesaux, D. (2010). Semi-heterarchical control of FMS: From theory to application. *Engineering Applications of Artificial Intelligence*, [ISSN: 09521976] *23*(8), 1314–1326. http://dx.doi.org/10.1016/j.engappai.2010.06.013.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, [ISSN: 03772217] *64*(2), 278–285. http://dx.doi.org/10.1016/0377-2217(93)90182-M.

Vespoli, Silvestro, Grassi, Andrea, Guizzi, Guido, & Santillo, Liberatina Carmela (2019). Evaluating the advantages of a novel decentralised scheduling approach in the Industry 4.0 and Cloud cenManufacturing era. *IFAC-PapersOnLine*, [ISSN: 24058963] *52*(13), 2170–2176. http://dx.doi.org/10.1016/j.ifacol.2019.11.527.

Zheng, Shuai, Gupta, Chetan, & Serita, Susumu (2020). Manufacturing dispatching using reinforcement and transfer learning. In *Machine learning and knowledge discovery in databases: European conference, ECML pKDD 2019, wüRzburg, Germany, September 16–20, 2019, proceedings, part III* (pp. 655–671). Springer.

Zimmermann, Emmanuel, Mezgebe, Tsegay Tesfay, Haouzi, Hind BRIL EL, Thomas, Philippe, Pannequin, Rémi, & Noyel, Melanie (2021). Multicriteria decision-making method for scheduling problem based on smart batches and their quality prediction capability. *Computers in Industry*, *133*, Article 103549.