Robust scheduling and dispatching rules for high-mix collaborative manufacturing systems

Andrea Maria Zanchettin[1]

Abstract

Motivated by the increasing demand of mass customisation in production systems, this paper proposes a robust and adaptive scheduling and dispatching method for high-mix human-robot collaborative manufacturing facilities. Scheduling and dispatching rules are derived to optimally track the desired production within the mix, while handling uncertainty in job processing times. The sequencing policy is dynamically adjusted by online forecasting the throughput of the facility as a function of the scheduling and dispatching rules. Numerical verification experiments confirm the possibility to accurately track highly variable production requests, despite the uncertainty of the system.

Keywords Dynamic resource scheduling and allocation · Simulation-based control of stochastic systems · High-mix production · Flexible manufacturing systems

1 Introduction

Agile manufacturing is a relatively new term adopted to describe a production approach able to respond quickly to unforeseen customer demands, market volatilities, or other factors of high manufacturing impact such as changing lot sizes, variants, process technologies. In contrast to lean manufacturing, one of the main principles of agile manufacturing is how to leverage the impact of production assets and data, while maintaining the lowest production costs. In such Flexible Manufacturing Systems (FMS) optimisation of the production capacity, as well as proper scheduling (see Pinedo (2012)) and dispatching strategies are paramount, see e.g. Shi et al. (2019); Ouelhadj and Petrovic (2009), and Blazewicz et al. (2019).

✉ Andrea Maria Zanchettin
  andreamaria.zanchettin@polimi.it

1 Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Via Ponzio 34/5, 20133 Milano, Italy

🖄 Springer

Due to the inherent complexity of the problem of dynamically allocating resources in the shop-floor, metaheuristics have been widely adopted in the literature as in Dörmer et al. (2015) and Saif et al. (2019). In particular, Genetic Algorithms (GA), see Filho et al. (2014); Yu et al. (2018); Bhosale and Pawar (2019), and Lin et al. (2019), have been very popular, especially in the last decade, to efficiently handle the combinatorial complexity of scheduling and dispatching problems. Gombolay et al. (2013) and Ivanov et al. (2016b) also report complete results, based on accurate mathematical modelling and optimisation algorithms. Stimulated by the novel reference paradigms of digital factories, the adoption of digital replicas of the shop-floor constantly updated with production analytics, i.e. the so-called *digital twins*, see Tao et al. (2018), are motivating the development of new approaches based on online simulation. Originally introduced in Wu and Wysk (1989), the possibility to online evaluate the current performance of the production facility using software-mediated data is a promising direction, Morel et al. (2003) and Wang et al. (2019). This possibility has been also exploited for the enhancement of both Manufacturing Operation Management (MOM) and Enterprise Resource Planning (ERP) systems, see Meyer et al. (2009) and Moon and Phatak (2005), respectively.

For a long time, automation has been an *all or nothing solution*, and automation has been a possible solution only for large OEM, especially in the automotive industry. Flexible as well as reconfigurable manufacturing systems are surely relevant in high-mix low volume production settings. On the other hand, their flexibility and reconfigurability are far from being enough in cases requiring flexibility comparable with craft production. Close cooperation between workers and partly automated assembly systems has been introduced as the best strategy to respond to the need for flexibility and quick changeability of manufacturing processes as described in Krüger et al. (2009), Karaulova et al. (2019) and Schlette et al. (2020). A new generation of industrial collaborative robots, those sharing their workspace with people without the need of safety fences, is allowing manufacturers to redesign their processes as suggested in Mateus et al. (2018), to achieve higher levels of efficiency as in Morioka and Sakakibara (2010). According to the work of Dianatfar et al. (2019), there are multiple trade-offs to be considered, between the high performance of dedicated robots, the great flexibility of human operators, and the compromise solution of collaborative robots. On the other hand, as Ding et al. (2014) and Messner et al. (2019) noticed, benefits arising from human-machine or human-robot collaboration are not offered without a significant increase in the complexity of scheduling and dispatching systems. More in detail, the synchronisation between humans and robots has to be handled by explicitly accounting for the stochasticity of the resulting workflow, as well as the reasons behind that, as described in Ferreira et al. (2018) and Ostermeier (2019), respectively. In order to properly handle the inherent stochasticity of collaborative production facilities, the combination of reactive and proactive/predictive methods, e.g. based on the estimation of the future performance, has to be further exploited as suggested in Cardin et al. (2017), or by using control strategies (see Ivanov et al. (2021) for an example and Ivanov et al. (2018) for an overview), like Model Predictive Control (MPC) as in Cataldo et al. (2015).

Make-to-order manufacturing processes are typically characterised by high-mix and low volume productions. In this context it might be beneficial to fully exploit

the flexibility of collaborative solutions by dynamically assign tasks to humans or robots in order to guarantee a continuous and sustained flow. On the other hand, the high variability of customers' requests, would make hardly possible to rely on a pre-prepared assignment table covering all the possible production mixes and volumes.

In summary, the simultaneous adoption of digital twins, possibly fed with advanced analytics coming from the field as in Li et al. (2015), and predictive approaches, seems to represent the cornerstone to robustly manage the execution of flexible and collaborative production layouts.

1.1 Novel contribution and related works

With the aim of gaining a competitive advantage in terms of speed to react to these market volatilities, companies are leveraging on advanced components and Information Technologies. By simulating the working conditions of operators, robots, and eventually of the whole plant, a scheduler can be sure that the real production will run as in- tended. In other words, a manufacturing scheduling system can leverage high-fidelity Digital Twins to accurately predict the future performance of the production facility by efficiently run several what-if analyses, and by simulating different scenarios to identify the best set of decisions to be applied on the physical plant.

This work presents a closed-loop control approach to robustly balance the time-varying high-mix production requirements in manufacturing systems. The high-mix multi-product scenario poses a series of challenges in finding an optimal scheduling solution, see e.g. Sprodowski et al. (2020). The balancing problem cannot be solved once and for all, as the mix is expected to change frequently. Frequent reallocations of resources are then foreseen and erroneous decisions can severely compromise the efficiency of the production system. The control strategy developed in this work relies on a forecasting technique based on a digital twin of the entire shop-floor. It selects the best scheduling and dispatching decisions to be executed, while robustly accounting for possible variabilities in the processing times. An outer control loop is finally responsible for tracking the desired production plans, typically coming from high-level management systems.

The main features of the algorithm are (1) the capability of handling a a time-varying mix in a multi-product scenario, (2) the robustness with respect to uncertainty, and (3) the adoption of a predictive strategy. The state-of-the-art of scheduling algorithms is rich of solutions. For example, Casalino et al. (2019) introduced an optimisation strategy to reduce the idle time (i.e. the amount non-value added activities), while dealing with uncertainty. Unfortunately, the method is not capable of handling multiple products. Prediction capabilities have been exploited in Cataldo et al. (2015), still for a single product, with limited contributions on robustness. Variable mix has been considered, in turn, Dörmer et al. (2015); Saif et al. (2019), and Wang et al. (2019), without characterising the expected robustness in face of uncertainty. The works in Ivanov et al. (2016a); Lin et al. (2019), and Shi et al. (2019), in turn, are specifically focused on robustness with respect to uncertain processing times. The present work differs from them mainly for the adoption of a predictive strategy, and the possibility to handle high-varying production mix.

In addition, the present work address the problem of scheduling by defining an optimisation problem that tries to minimise the discrepancy of the actual production from a reference one (i.e. the tracking error), instead of maximising the throughput or, equivalently, minimising the idle time as suggested in other works. In summary, Table 1 reports a comparison of the analysed works, based on the three aforementioned features proposed in this work. As one can notice, none of the existing work is apparently capable of simultaneously handle the three main characteristics of the algorithm described in the following. Moreover, looking at the main objectives of the listed methods, the method proposed in this work is the sole introducing the means squared value of the production error as a cost function to be optimised (see again Table 1).

1.2 Structure of the paper

The reminder of the paper is structured as follows. Section 2 depicts the overall architecture of the system, and describes the modelling principles of Discrete EVent Systems (DEVS) specification. Section 3 introduces the optimal scheduling algorithm, while Sect. 4 contains further details regarding the resolution of the optimisation problem that will eventually select the best policy to be applied. Finally, Sect. 5 presents the outcome of a verification scenario, consisting in two multiple-product assembly lines and discusses the outcome of the numerical validation.

2 Modelling principles

This Section sketches the overall picture of the developed method, as well as the modelling principles adopted throughout the paper. The list of the adopted symbols and their meanings is given in the Appendix.

The scheduling and despatching algorithm is mainly composed of three functional blocks:

1. the *Plant Digital Twin* contains a replica of the *Manufacturing Plant*; its state is updated based on *Events* coming from the plant, whilst its parameters are identified from *Analytics* (i.e. processing times of jobs collected from previous executions) obtained from an *Events Logger* which is, in turn, stored within an Enterprise Resource Planning (ERP) system;
2. the *Throughput controller* exploits the *Plant Digital Twin* to forecast the production of the manufacturing system as a function of the possible dispatching and the scheduling decisions and is responsible for selecting the best activities to execute, i.e. the input $u(t)$ to the plant;
3. the *Production controller* finally compares, for all products $p \in \mathbb{P}$, the actual production $\theta_p(t)$ with the desired one $\theta_p^0(t)$ and computes the reference throughput $\omega_p^0(t)$.

Table 1 Summary of the comparison with existing methods

Paper	Variable mix	Robustness wrt. uncertainty	Predictive	Cost function
Casalino et al. (2019)		✓	✓	Iidle time
Cataldo et al. (2015)			✓	Multi-objective (production, energy, etc.)
Dörmer et al. (2015)	✓			Balancing / maximum workload
Ivanov et al. (2016a)		✓		Multi-objective (throughput, makespan, lateness, etc.)
Lin et al. (2019)		✓		Makespan
Saif et al. (2019)	✓			Multi-objective (makespan, material, lateness)
Shi et al. (2019)		✓		Flowtime
Wang et al. (2019)	✓	✓		Multi-objective (processing time and logistics costs)
Proposed method	✓	✓	✓	Mean squared value of production tracking error

Figure 1 depicts the overall architecture of the proposed system and details the interconnections between the three components.

In the following, modelling principles for the *Plant Digital Twin* are given, while details regarding both the *Production controller* and the *Throughput controller* are given in the following Section.

Without lack of generality, the behaviour of the *Manufacturing plant* as well as its replica within the *Plant Digital Twin* will be described in terms of DEVS specification, see e.g. Zeigler et al. (2000); Tendeloo and Vangheluwe (2018). Other formalisms, like pcTPN (partially controllable Time Petri Nets) Ramadge and Wonham (1987) or any other paradigm, see e.g. Cassandras and Lafortune (2009), can be equivalently adopted.

Based on the definition of DEVS proposed in Zeigler et al. (2000), we here define a partially controllable DEVS (pcDEVS) to represent the *Plant Digital Twing* as the model expressed by the following tuple:

$$\text{pcDEVS} = \langle \mathcal{U}, \mathcal{S}, ta, \delta_{int}, \delta_{ext} \rangle$$

where $\mathcal{U}$ is the set of inputs and represents all the possible scheduling commands, while $\mathcal{S}$ represents the possible states of the system. The set of states is partitioned into $\mathcal{S}_{int}$ and $\mathcal{S}_{ext}$ (i.e. $\mathcal{S} = \mathcal{S}_{int} \cup \mathcal{S}_{ext}, \mathcal{S}_{int} \cap \mathcal{S}_{ext} = \emptyset$). A state $\bar{s} \in \mathcal{S}_{ext}$ represents the situation when a scheduling command in $\mathcal{U}$ can be selected to steer the evolution of the system, which happens, for example, when a certain job is completed. In turn, a state $s \in \mathcal{S}_{int}$ stands for the case when no control input can be selected, i.e. when all the agents are occupied in processing their jobs and no command can be executed. Function $ta : \mathcal{S}_{int} \to \mathbb{R}_0^+ = [0, +\infty)$ stands for the time advance function and determines the lifespan of an internal state. Finally $\delta_{int} : \mathcal{S}_{int} \times \mathbb{R}_0^+ \to \mathcal{S}$ and $\delta_{ext} : \mathcal{S}_{ext} \times \mathcal{U} \to \mathcal{S}$ stand for the internal and the external transition functions, respectively. The former identifies autonomous timed state transitions, while the latter identifies immediate command-driven state transitions.

As an example, Fig. 2 exemplifies the case of a pair of machineries. State $\bar{s}_1 \in \mathcal{S}_{ext}$ represents the first machine being idle, while the second machine is processing a part. As the first machine is able to operate a new part, a control command, u_1 can be used to start process a new job, triggering a state change from $\bar{s}_1$ to $s_2 \in \mathcal{S}_{int}$ (external, i.e.
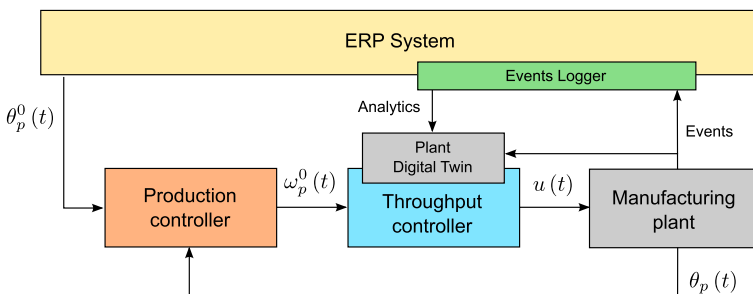

Fig. 1 Architecture of the developed method and its three main components: the *Production controller*, the *Throughput controller*, and the *Plant Digital Twin*
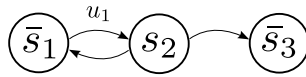
Fig. 2 A partially controllable Discrete EVent System (pcDEVS)

controllable, state transition). In state s_2 no control command is available as the two machines are both operating. When the system is in this state, two possible events can occur. In case the first machine finishes its job before the second one, the state changes to $\bar{s}_1$ (an internal, uncontrollable, transition). In turn, if the second machine finishes earlier than the other, that state will be updated to $\bar{s}_3$. In this state, decision on the job to assign to the second machine can be taken. Consistently the state $\bar{s}_3$ is an external one, i.e. $\bar{s}_3 \in \mathcal{S}_{ext}$.

When the evolution of a pcDEVS is seen only within $\mathcal{S}_{ext}$, the corresponding behaviour can be equivalently described by a Markov Decision Process (MDP). For the same example, Fig. 3 reports the equivalent MDP. From state $\bar{s}_1$, the control input u_1 can bring the system to either state $\bar{s}_3$ or to state $\bar{s}_1$. The former transition (from $\bar{s}_1$ to $\bar{s}_3$) will happen in case the first machine will finish the job before the other, while the latter ($\bar{s}_1$ to $\bar{s}_1$) will happen in case the operation of second machine will be concluded before the one assigned to the first machine.

Summarising, the *Plant Digital Twin* of a flexible manufacturing process can be characterised by a MDP. Its states $\bar{s} \in \mathcal{S}_{ext}$ are conditions of the production process corresponding to one or more agents available to start processing new jobs. Therefore, if a scheduling command $u \in \mathcal{U}$ is selected in certain state of the MDP, the system can evolve stochastically to a new state.

3 Development of the scheduling algorithm

A scheduling and dispatching policy π is a function $\pi : \mathcal{S}_{ext} \to \mathcal{U}$ that maps the actual state $\bar{u} \in \mathcal{S}_{ext}$ to a scheduling and dispatching decision $u \in \mathcal{U}$. This Section defines the optimality principle for finding the best scheduling and dispatching policy π^* that, for each product p within the mix $\mathbb{P}$, will ensure the minimum deviation from the desired throughput. Assume that the scheduler is requested to operate at current time instant t, given the desired throughput $\omega_p^0(t)$. Then the quantity $\omega_p^0(t)(\tau - t)$ represents the desired production of products of type p, i.e. the number of products of type p to be produced in the interval $[t, \tau]$, and $\widehat{\Delta\theta_{p,\pi}}(\tau) = \widehat{\theta_{p,\pi}}(\tau) - \theta_p(t)$ the desired increment in the number of available products of the same type, i.e. the ones that will be actually produced under policy π during the same interval, the quantity

$$\omega_p^0(t)(\tau - t) - \widehat{\Delta\theta_{p,\pi}}(\tau)$$

Fig. 3 Markov Decision Process (MDP) equivalent to the pcDEVS of Fig. 2
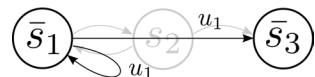
stands for the production error at time instant $\tau \in [t, t + \Gamma]$ for product $p \in \mathbb{P}$, see Fig. 4.

This quantity clearly depends on the policy π, which in turn returns the scheduling and dispatching commands, as well as on the processing times of all activities involved in the production of p. In order to account for the future evolution of the manufacturing process, the squared value of the production error is integrated along the prediction horizon, i.e.

$$\frac{1}{\Gamma} \int_t^{t+\Gamma} \left(\omega_p^0(t)(\tau - t) - \widehat{\Delta \theta_{p,\pi}}(\tau) \right)^2 d\tau \tag{1}$$

thus to require the mean square production error to be as small as possible. Since the processing times, and therefore the lifespan of each state $s \in \mathcal{S}_{int}$, are regarded as stochastic variables, the mean square production error is a stochastic variable as well. In order to robustly select the optimal scheduling and dispatching policy π^*, the following optimisation problem is finally introduced:

$$\pi^* = \arg\min_\pi \mathbb{E}_\pi [J_\pi] \tag{2}$$

where

$$J_\pi = \sum_{p \in \mathbb{P}} \frac{1}{\Gamma} \int_t^{t+\Gamma} \left(\omega_p^0(t)(\tau - t) - \widehat{\Delta \theta_{p,\pi}}(\tau) \right)^2 d\tau \tag{3}$$

The optimisation problem in (2) attempts to find optimal set of scheduling and dispatching policy π^* at present time instant t by minimising, for all products $p \in \mathbb{P}$, the conditional expectation $\mathbb{E}_\pi[\cdot]$, under π, of the mean square production error, once the reference throughputs $\omega_p^0(t)$, one per each product in the mix, are given. Notice, that the *Throughput controller* is exploiting the *Plant Digital Twin* of the manufacturing plant for the prediction of the increment in the number of available product of type p as a function of the scheduling and dispatching policy π, i.e. $\widehat{\Delta \theta_{p,\pi}}(\tau)$. For
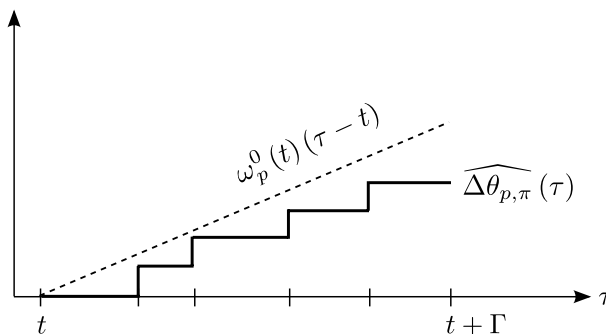
Fig. 4 Typical result of a simulation in terms of predicted increment in the number of available product of type p, $\widehat{\Delta \theta_{p,\pi}}(\tau)$, and corresponding reference value $\omega_p^0(t)(\tau - t)$ - vertical ticks represent time instants corresponding to production events of product p

this reason, the *Plant Digital Twin* is constantly updated with the last available production analytics retrieved from the *ERP System* (see Fig. 1).

At the present time instant t, the desired throughput $\omega_p^0(t)$ in (2) is computed so to be proportional to the actual production error $\theta_p^0(t) - \theta_p(t)$:

$$\omega_p^0(t) = \begin{cases} K\left(\theta_p^0(t) - \theta_p(t)\right) & \theta_p^0(t) > \theta_p(t) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

thus implementing the actual closed-loop production control law, where $K > 0$ represents the proportional gain. Differently from the *Throughput controller* that exploits the digital replica of the manufacturing plant for prediction purposes, the *Production controller* in (4) implements a closed-loop regulation strategy using actual production data coming from the manufacturing plant. As commanding a negative throughput has clearly no physical meaning, when the actual production exceeds the desired one (i.e. when $\theta_p(t) > \theta_p^0(t)$), the corresponding reference throughput $\omega_p^0(t)$ is set to zero, so to temporarily stop the production of p.

Finally, the desired number of product of type p at time instant t, $\theta_p^0(t)$, is assumed to be computed based on the orders of product p stored in the *ERP System* as follows:

$$\theta_p^0(t) = \left\lceil \sum_{i_p} \begin{cases} 0 & t < t_{i,p} \\ Q_{i,p}\frac{t - t_{i,p}}{\Delta T_{i,p}} & t_{i,p} \le t \le t_{i,p} + \Delta T_{i,p} \\ Q_{i,p} & \text{otherwise} \end{cases} \right\rceil \tag{5}$$

where $\Delta T_{i,p}$ is the takt-time associated to the i-th order of product p, $t_{i,p}$ is the corresponding order arrival time, $Q_{i,p}$ is the ordered quantity, while finally $\lceil \cdot \rceil$ denotes the *ceil* function. An example of the application of Eq. (5) is reported in Fig. 5.

Optionally, the following feedforward term $\dot{\theta}_p^0(t)$ can be included:
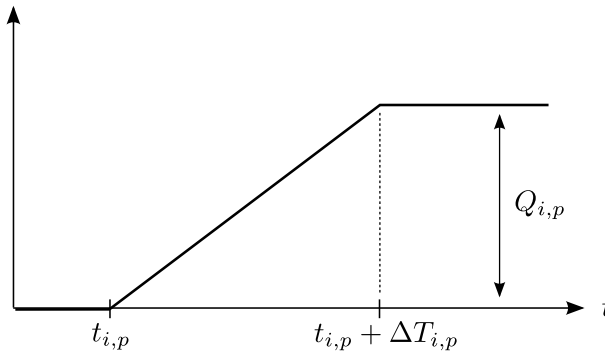
Fig. 5 Graphical explanation of Eq. (5), at time $t_{i,p}$ an order to produce $Q_{i,p}$ products in $\Delta T_{i,p}$ arrives. The corresponding desired number of product of type p to be available is increased to account for the newly arrived order. The increase is applied gradually until the desired quantity $Q_{i,p}$ is reached at the requested delivery time $t_{i,p} + \Delta T_{i,p}$

$$\dot{\theta}_p^0(t) = \sum_{i_p} \begin{cases} \dfrac{Q_{i_p}}{\Delta T_{i,p}} & t_{i_p} \le t \le t_{i_p} + \Delta T_{i,p} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Consistently, the reference throughput in (4) has to be redefined as follows:

$$\omega_p^0(t) = \begin{cases} \dot{\theta}_p^0(t) + K\left(\theta_p^0(t) - \theta_p(t)\right) & \theta_p^0(t) > \theta_p(t) \\ 0 & \text{otherwise} \end{cases}$$

So far, a high-level picture of the scheduling method has been sketched. The role of simulations performed on the *Plant Digital Twin* as well as their stochastic nature has been described. In the next Section, further details will be given on how to actually solve the scheduling problem that will eventually decide which action should be executed next, in order to guarantee the reference production, despite the possible variabilities in the processing times.

4 Optimisation algorithm

In the following, the algorithm adopted to solve the optimisation problem in (2) is further detailed. A detailed and formal discussion on the robustness of the algorithm follows at the end of this section. Given the definition introduced in the previous Section, it is now possible to introduce Algorithm 1 that computes the cost c associated to a given trajectory. Throughout the reminder of the manuscript, the term trajectory $\{e_i\} = \{e_1, e_2, e_3, \dots\}$ is used to identify the sequence of states and corresponding timestamps, i.e. $e_i = \langle \bar{s}_i, \tau_i \rangle$ identifies a certain pair $\mathcal{S} \times \tau$. For a given trajectory, Algorithm 1 simply evaluates the corresponding cost in (3).

Algorithm 1 Cost calculation for trajectory $\{e_i\}$

1: **procedure** CALCULATECOST($t, \Gamma, \{\omega_p^0(t)\}, \{e_i\}$)
2:     $C \leftarrow 0$;
3:     $c \leftarrow 0$;
4:     **for all** $p \in \mathbb{P}$ **do**
5:         $A \leftarrow 0$;
6:         $B \leftarrow 0$;
7:         $\tau_0 \leftarrow t$;
8:         **for all** $e_j = \langle s_j, \tau_j \rangle \in \{e_i\}$ **do**
9:             **if** s_j corresponds to the production of p **then**
10:                 $A \leftarrow A + C^2 (\tau_j - \tau_0)$;
11:                 $B \leftarrow B + C\left[(\tau_j - t)^2 - (\tau_0 - t)^2\right]$;
12:                 $C \leftarrow C + 1$;
13:                 $\tau_0 \leftarrow \tau_j$;
14:         $A \leftarrow A + C^2 (\Gamma - \tau_0)$;
15:         $B \leftarrow B + C\left[\Gamma^2 - (\tau_0 - t)^2\right]$;
16:         $c \leftarrow c + \Gamma \dfrac{\left(\omega_p^0(t)\, \Gamma\right)^2}{3} + A - \omega_p^0(t)\, B$;
17:     **return** c/Γ;

As stated previously, due to the stochastic nature of the *Plant Digital Twin*, the trajectory $\{e_i\}$ is a stochastic process, and therefore the associated cost c is a stochastic variable. Therefore, in order to attain a robust decision regarding the scheduling and the dispatching rules, a high number N of trajectories are generated by the *Plant Digital Twin* using a Monte Carlo sampling. The outcome of these simulations is in terms of the dataset $D = \{d_j\}, d_j = \langle \{s_i\}_j, J_{\pi_j} \rangle$, i.e. a sequence of states together with the corresponding cost J_{π_j}. In order to minimise the expected cost in (2), and therefore to evaluate the best scheduling and dispatching policy π^* and the corresponding command(s) $u(t)$ to be used at the present time instant t, an agglomerative (i.e. bottom-up) hierarchical clustering method of the data set D is adopted.

Assume that N sampled trajectories are available, each in its own cluster with the corresponding cost, and assume that a subset of them only differs in the last state. If the last states are internal ones, i.e. $s \in \mathcal{S}_{int}$, the corresponding clusters are merged together, moving up the hierarchy. The cost associated to this new bigger cluster is composed by the union of the costs of the trajectories in the subset, and the last states are cancelled. For example, consider at iteration k the two following clusters:

$$C_1^{(k)} = \langle\langle 0, 1, 11, 2, \cdots, 3, 5, 3\rangle, 31.6\rangle$$
$$C_2^{(k)} = \langle\langle 0, 1, 11, 2, \cdots, 3, 5, 7\rangle, 34.6\rangle$$

Assume that the last state are internal ones, i.e. $3, 7 \in \mathcal{S}_{int}$. Then, at iteration $k+1$ a new cluster is generated containing the common part of the two trajectories and the union of the costs, i.e.

$$C_1^{(k+1)} = \emptyset$$
$$C_2^{(k+1)} = \langle\langle 0, 1, 11, 2, \cdots, 3, 5\rangle, \langle 31.6, 34.6\rangle\rangle$$

Costs are merged together as the corresponding states are internal one, and cannot be controlled using the input command.

In turn, if the last states are external ones, i.e. $s \in \mathcal{S}_{ext}$, the cluster with higher expected cost will be simply deleted (i.e. not propagated to the next iteration), and the last states are cancelled. Consider for example:

$$C_2^{(k+1)} = \langle\langle 0, 1, 11, 2, \cdots, 3, 5\rangle, \langle 31.6, 34.6\rangle\rangle$$
$$C_3^{(k+1)} = \langle\langle 0, 1, 11, 2, \cdots, 3, 9\rangle, \langle 35.1\rangle\rangle$$

where $5, 9 \in \mathcal{S}_{ext}$, then at the next iteration $k+2$, the following clusters will be considered

$$C_2^{(k+2)} = \langle\langle 0, 1, 11, 2, \cdots, 3\rangle, \langle 31.6, 34.6\rangle\rangle$$
$$C_3^{(k+2)} = \emptyset$$

This procedure will continue, until only one cluster remains that will represent the optimal command, in the sense of the cost function in (2). This action corresponds to a possible scheduling decision that will influence the future behaviour of the

manufacturing process. As the state is an external one, and therefore it depends on the policy π, the clustering algorithm neglects future evolution of the manufacturing process with higher costs as they can be avoided by an alternative selection the scheduling policy. The overall procedure is formally explained in Algorithm 2.

Algorithm 2 Agglomerative Trajectory Clustering

1: **procedure** CLUSTERING(D)
2:     $k \leftarrow 0$;
3:     **for** $i = 1$ **to** $i = N$ **do**
4:         $C_i^{(k)} \leftarrow \langle \{s_i\}_j, J_{\pi_i} \rangle$;
5:     $F \leftarrow$ true;
6:     **while** F **do**
7:         $F \leftarrow$ false;
8:         **for each** $i \neq j : C_i^{(k)}.s_{1:e-1} = C_j^{(k)}.s_{1:e-1}$ **do**
9:             $F \leftarrow$ true;
10:             **if** $s_e \in \mathcal{S}_{ext}$ **then**
11:                 **if** $\hat{\mathbb{E}}\left[C_i^{(k)}.J_{\pi_i}\right] > \hat{\mathbb{E}}\left[C_j^{(k)}.J_{\pi_j}\right]$ **then**
12:                     $C_i^{(k+1)} \leftarrow \emptyset$;
13:                     $C_j^{(k+1)} \leftarrow \langle C_j^{(k)}.s_{1:e-1}, C_j^{(k)}.J_{\pi_j} \rangle$;
14:                 **else**
15:                     $C_i^{(k+1)} \leftarrow \langle C_i^{(k)}.s_{1:e-1}, C_i^{(k)}.J_{\pi_i} \rangle$;
16:                     $C_j^{(k+1)} \leftarrow \emptyset$;
17:             **else**
18:                 $C_i^{(k+1)} \leftarrow \emptyset$;
19:                 $C_j^{(k+1)}.s \leftarrow C_j^{(k)}.s_{1:e-1}$;
20:                 $C_j^{(k+1)}.c \leftarrow C_i^{(k)}.J_{\pi_i} \cup C_j^{(k)}.J_{\pi_j}$;
21:     $k \leftarrow k + 1$;

At the end of the iterative procedure in Algorithm 2, the unique remaining cluster will contain the optimal external state s happening at present time instant t that, for all products p in the mix $\mathbb{P}$, will ensure, at least for $N \to \infty$, a robust and optimal tracking of the reference throughput $\omega_p^0(t)$ within the prediction horizon. The same algorithm will be invoked cyclically every time a new scheduling or dispatching rule can be applied.

The approach described in Algorithm 2 is similar to the one proposed in Casalino et al. (2019), but here adapted to a different cost function and a generic modelling formalism. In particular, the work in Casalino et al. (2019) introduced an algorithm to maximise the throughput of the production of a single product, while here the objective is rather to minimise the production error in high-mix productions.

4.1 Notes on the robustness of the algorithm

Based on the described algorithm, it is worth understanding its robustness, i.e. how the probability of actually selecting the best policy is related to how simulations are distributed within the space of available policies. The Monte Carlo strategy will uniformly explore the space of possible decision policies. Assume that a certain policy π_i has been evaluated $N_i \geq 1$ times, and that the corresponding sampled costs are $J_{\pi_i}^{(k)}$, $k = 1, \ldots, N_i$. Then, as anticipated, the expected cost of each policy $\mu_i = \mathbb{E}\left[J_{\pi_i}\right]$ will

be evaluated through its empirical mean, i.e. $\hat{\mathbb{E}}[J_{\pi_i}] = 1/N_i \sum_{k=1}^{N_i} J_{\pi_i}^{(k)}$ (see Algorithm 2, line 11). The same applies for another policy π_j, $i \neq j$. Define $\mathcal{P}_0$ the probability to prefer policy π_i to π_j, given that π_i guarantees better performance than π_j, i.e. $\mu_i < \mu_j$, in terms of the cost function in (3). Such a probability, that clearly indicates how effective the scheduler is in recognising that policy π_i outperforms π_j, can be formally defined as follows

$$\mathcal{P}_0 = \mathcal{P}\left(\hat{\mathbb{E}}[J_{\pi_i}] < \hat{\mathbb{E}}[J_{\pi_j}] \middle| \mu_i < \mu_j \right)$$

which stands for the probability to take the correct decision in Algorithm 2, line 11. Assume that the two distributions J_{π_i} and J_{π_j} have variances $\mathbb{V}\mathrm{ar}[J_{\pi_i}] = \sigma_i^2$ and $\mathbb{V}\mathrm{ar}[J_{\pi_j}] = \sigma_j^2$, respectively. Therefore the two empirical means $\hat{\mathbb{E}}[J_{\pi_i}]$ and $\hat{\mathbb{E}}[J_{\pi_j}]$ have means μ_i and μ_j, respectively, and variances

$$\mathbb{V}\mathrm{ar}\left[\hat{\mathbb{E}}[J_{\pi_{(i,j)}}]\right] = \frac{\mathbb{V}\mathrm{ar}[J_{\pi_{(i,j)}}]}{N_{(i,j)}} = \frac{\sigma_{(i,j)}^2}{N_{(i,j)}}$$

Then, the quantity $\hat{\mathbb{E}}[J_{\pi_i}] - \hat{\mathbb{E}}[J_{\pi_j}]$ has mean $\mu_i - \mu_j$ (which is negative under the assumption $\mu_i < \mu_j$) and variance

$$\sigma^2 = \mathbb{V}\mathrm{ar}\left[\hat{\mathbb{E}}[J_{\pi_i}]\right] + \mathbb{V}\mathrm{ar}\left[\hat{\mathbb{E}}[J_{\pi_j}]\right] - 2\mathbb{C}\mathrm{ov}\left[J_{\pi_i}, J_{\pi_j}\right]$$

Assume that the same quantity is normally distributed, i.e.

$$\hat{\mathbb{E}}[J_{\pi_i}] - \hat{\mathbb{E}}[J_{\pi_j}] \sim \mathcal{N}(\mu_i - \mu_j, \sigma^2)$$

Then, probability $\mathcal{P}_0$ can be evaluated as follows:

$$\mathcal{P}_0 = \mathcal{P}\left(\mathcal{N}(0,1) < \frac{\mu_j - \mu_i}{\sigma}\right)$$

where

$$\sigma = \sqrt{\frac{\sigma_i^2}{N_i} + \frac{\sigma_j^2}{N_j} - 2\mathbb{C}\mathrm{ov}\left[J_{\pi_i}, J_{\pi_j}\right]}$$

Defining $\eta = N_i/N_j$, the probability $\mathcal{P}_0$ to prefer policy π_i to π_j, given that π_i guarantees better performance than π_j according to (3), is finally given by

$$\mathcal{P}_0 = \mathcal{P}(\mathcal{N}(0,1) < k)$$

where k, which is positive under the assumption $\mu_i < \mu_j$, is defined as follows

$$k = \frac{\sqrt{N_i + N_j}\left(\mu_j - \mu_i\right)}{\sqrt{\left(1 + \eta^{-1}\right)\sigma_i^2 + (1 + \eta)\sigma_j^2 - 2\sqrt{N_i + N_j}\,\mathbb{C}\text{ ov}\left[J_{\pi_i}, J_{\pi_j}\right]}}$$

For given values of $\mu_i, \mu_j, \mathbb{C}\text{ ov}\left[J_{\pi_i}, J_{\pi_j}\right]$ and $N_i + N_j$, the probability $\mathcal{P}_0$ has its maximum for $\eta = \sigma_i/\sigma_j$, and therefore the Monte Carlo approach that uniformly explores the space of available policies, i.e. $\eta \approx 1$, corresponds to the most promising method, at least when $\sigma_i \approx \sigma_j$. Without increasing the overall number of samples $N \geq N_i + N_j$, the sole way to improve the performance of the optimisation algorithm in selecting the best policy is to maximise $\mathbb{C}\text{ ov}\left[J_{\pi_i}, J_{\pi_j}\right]$, resorting to variance reduction techniques, such as Common Random Numbers (CRN), see Glasserman and Yao (1992). More specifically, with the aim of maximising k and hence $\mathcal{P}_0$, samples from the same pool of possible durations will be sampled for each possible activity and consistently used within all of the N Monte Carlo simulations.

5 Verification

In order to verify the presented approach, a validation scenario consisting in a hybrid human-robot assembly plant is assumed.

5.1 Verification scenario

The manufacturing layout (see Fig. 6) allows to assemble 3 variants of the same product, i.e. $\mathbb{P} = \{\#1, \#2, \#3\}$.

Two parallel lines are arranged, the first (top of Fig. 6) for product #1 and #2, the other (bottom) for products #2 and #3. Product #2, that is assumed the one having the highest demand, can be produced in parallel on the two lines. Each line is divided in three nearly balanced jobs, each performed in a dedicated station: stations #A, #B and #C for the first line, #D, #E and #F for the second. The workflow follows the alphabetical order (i.e. from #A to #C and from #D to #E, see again Fig. 6). Between each pair of consecutive stations, two buffers with limited capacity are placed to temporarily store the work-in-progress (WiP).

Two robots on mobile bases and two human operators are allowed to move along the two lines and to occupy different stations, so as to maximise the flexibility. Operations within stations #A and #D can only be performed by human workers, while those in stations #C and #F are for robots only. The others (those in stations #B and #E) can be performed by either one operator, or one robot, thought with different level of efficiency. Similarly, operations performed by robots are also characterised by different cycle times, depending on the robot allocated to the corresponding task. Table 2 reports the processing times, both in case of manual execution or robotic execution. As for simulation purposes, distributions of the processing times are assumed to be Gaussian (Table 2 reports the standard deviations, as well). Finally,

in case the requested production will result to be significantly lower than the maximum capacity of the production plant, a fictitious activity has been added, that corresponds to an idle time. In order to give the reader a taste regarding the combinatorial complexity of the scheduling and dispatching problem, the total number of possible commands is 96, and millions of possible states. This number identifies the maximum number of possible decisions for the scheduler at each iteration.

5.2 Simulation and outcomes

Thirty minutes (1800 seconds) of manufacturing execution have been numerically simulated on a 2.9 *GHz* INTEL CORE I5, with 16 *GB* 1867 *MHz* DDR3 of memory. The *N* simulations are performed independently exploiting the multithreading functionalities of the CPU. Reported execution times for the whole procedure are of 475 ± 49 milliseconds. The list of orders, with arbitrarily generated arrival times, are shown in Table 3, together with the corresponding quantity and due dates, and are stored within the *ERP System*. The orders are accessible via mySQL queries that are performed at each cycle of the scheduling algorithm. The references $\theta_p^0(t)$, computed according to (5), are reported in Fig. 7. Significant fluctuation of the takt-times and of the overall production request can be appreciated. Finally, the parameters of the control strategy adopted within the simulation experiments are reported in Table 4.

At each iteration, the scheduler executes a query on the mySQL database to retrieve the updated list of orders and, for each product, updates the corresponding reference production $\theta_p^0(t)$ using (5) as well as the reference throughput $\omega_p^0(t)$ according to (4). Then, it performs *N* Monte Carlo simulations on the *Plant Digital Twin*, whose parameters are retrieved from the mySQL database, through runtime analytics. Each simulation is then evaluated in terms of the corresponding cost in (3) by applying Algorithm 1. The whole set of simulations and corresponding costs are clustered according to Algorithm 2 and the corresponding optimal and robust dispatching and scheduling policy π^*, i.e. the one satisfying the optimality of (2), is obtained.

The outcome of the experiment is shown in Fig. 9, which reports the actual production $\theta_p(t)$ as well as the Gantt chart that is incrementally generated by the scheduling algorithm. As the reader can see, the manufacturing facility is able to adjust its pace as well as the resource allocation strategy based on the actual demand (see again Table 3). For example, since the request for products of type #3 increases in the last 5 minutes, the scheduler correctly decides to allocate more resource on the corresponding jobs. Moreover, it is worth to mention that these capabilities are robustly achieved with respect to the variability of processing times, see again Table 2.

5.3 Analysis of robustness

Finally, in order to evaluate the robustness of the optimisation algorithm, which has been formally discussed in Subsection 4.1, eight additional executions of the same

Table 2 Durations of the activities (and corresponding standard deviations), including waiting and transfer from one station to another (values in seconds)

Agent	#A		#B		#C		#D		#E		#F		Idle
	#1	#2	#1	#2	#1	#2	#2	#3	#2	#3	#2	#3	
Robot #1			10 ± 5%	15 ± 5%	11 ± 5%	14 ± 5%			15 ± 5%	19 ± 5%	16 ± 5%	19 ± 5%	5
Robot #2			9 ± 5%	16 ± 5%	9 ± 5%	16 ± 5%			14 ± 5%	21 ± 5%	15 ± 5%	21 ± 5%	5
Operators	10 ± 10%	16 ± 10%	12 ± 10%	16 ± 10 %			16 ± 10%	21 ± 10%	16 ± 10%	21 ± 10%			5

Agent		#A	#B	#C	#D	#E	#F
Robots	#A						
Robots	#B			3 ± 3%		5 ± 3%	7 ± 3%
Robots	#C		3 ± 3%			7 ± 3%	5 ± 3%
Robots	#D						
Robots	#E		5 ± 3%	7 ± 3%			3 ± 3%
Robots	#F		7 ± 3%	5 ± 3%		3 ± 3%	
Operators	#A		2 ± 5%		3 ± 5%	4 ± 5%	
Operators	#B	2 ± 5%			4 ± 5%	3 ± 5%	
Operators	#C						
Operators	#D	3 ± 5%	4 ± 5%			2 ± 5%	
Operators	#E	4 ± 5%	3 ± 5%		2 ± 5%		
Operators	#F						
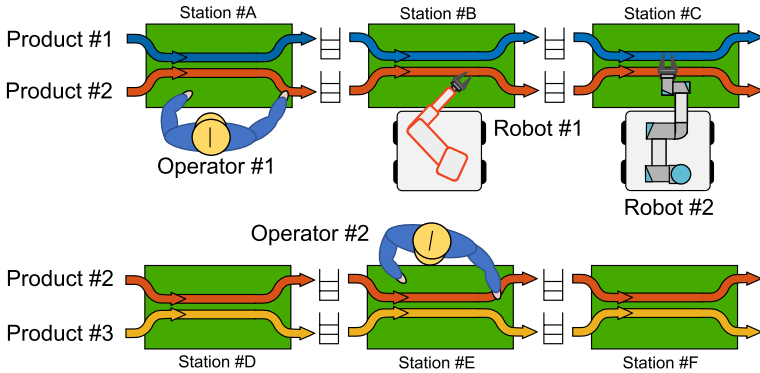
Fig. 6 Layout for the validation scenario composed of 2 lines with 3 stations to assemble $N_{\mathbb{p}} = 3$ variants of the same product

Table 3 List of orders

Order ID	Arrival time	Deliver in (min)	Product	Qty
#1	2019-12-01 12:00:00	20	#1	12
#2	2019-12-01 12:00:00	20	#2	19
#3	2019-12-01 12:10:00	10	#3	2
#4	2019-12-01 12:10:00	20	#2	14
#5	2019-12-01 12:15:00	10	#1	6
#6	2019-12-01 12:25:00	5	#3	11

scenario has been collected, using the same parameters listed in Table 4. The corresponding behaviours are reported in Fig. 8.

Several Key Performance Indicator (KPIs) have been introduced for the evaluation of the corresponding output. Table 5 reports the average lateness for each order together with the corresponding standard deviation. It is worth noticing that the algorithm tends to anticipate the production (i.e. to have a negative lateness). This fact is surely related to the contribution $\dot{\theta}_p^0$ given by the feedforward action to the reference throughput ω_p^0, as defined in (6).

With reference to Fig. 8, for each product p in the mix, the root means square error (RMSE), defined as

$$RMSE_p = \sqrt{\frac{1}{1800} \int_0^{1800} \left(\theta_p^0(\tau) - \theta_p(\tau) \right)^2 d\tau}$$

which quantifies the variability of production with respect to the desired value, has an average of 0.3522 parts (for product #1), 0.5698 (# 2), and 0.5620 (product # 3).
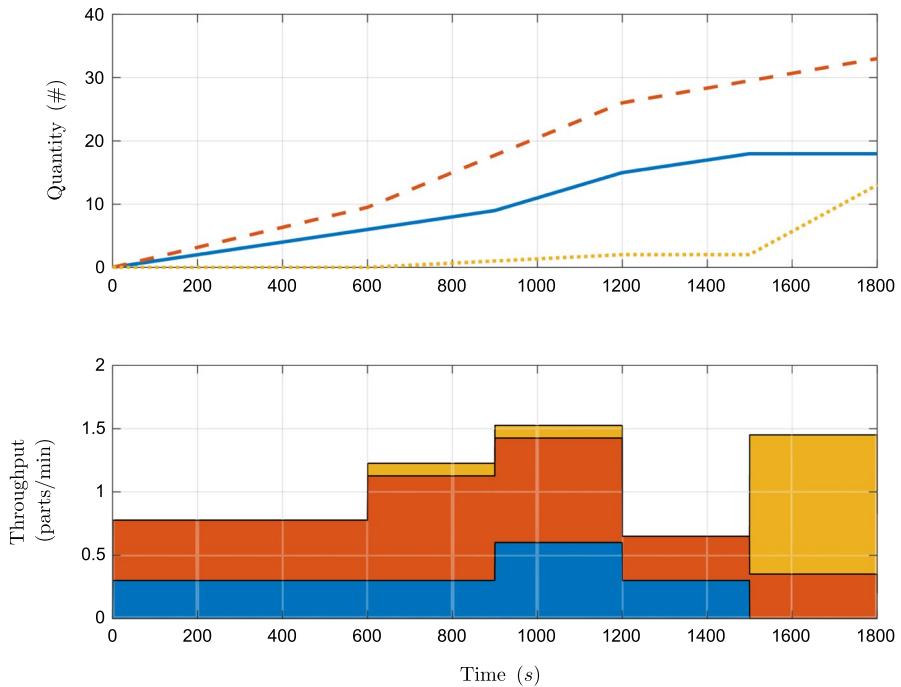
Fig. 7 Reference production for the 3 products (#1: solid blue, #2: dashed red, and #3: dotted yellow, respectively) with exception for the *ceil* function (top), and corresponding throughput (bottom) (Color figure online)

Table 4 Parameters for the verification scenario

Parameter	Description	Value
N	Number of simulations	1000
Γ	Prediction horizon	120 s
K	Controller gain	0.5 Hz
f_s	Scheduling frequency	$\sim 2.0\ Hz$

In turn, the average span, evaluated as the value of the maximum variability of the production (i.e. the thickness of the shaded areas in Fig. 8) has been evaluated as 0.3728 parts (for product #1), 1.0711 (product #2), and 0.3378 (#3).

Overall, as expected, the proposed scheduling strategy is able to effectively mitigate the variability of the durations of the different activities and to produce good tracking capabilities of the desired production θ_p^0, even in case of a highly variable mix (see Fig. 7).
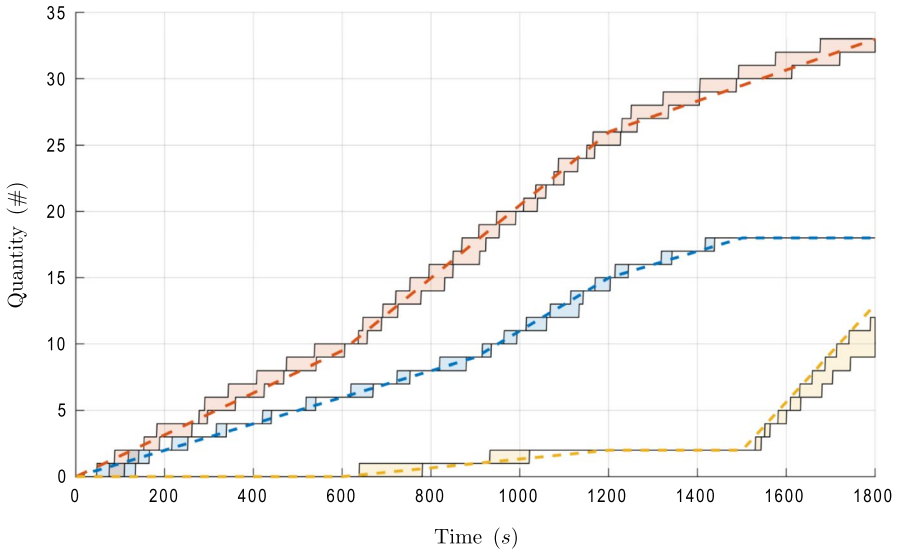
Fig. 8 Variability of the actual production in 8 simulations (shaded areas) and the corresponding reference (dashed)

Table 5 List of orders, actual delivery lateness (tardiness, if positive, earliness, otherwise) averaged among the 8 runs, and corresponding standard deviations

Order ID	Deliver in (min)	Avg. lateness (s)	Std. lateness (s)
#1	20	7	27
#2	20	8	26
#3	10	−233	26
#4	20	−50	35
#5	10	−76	7
#6	5	43	27
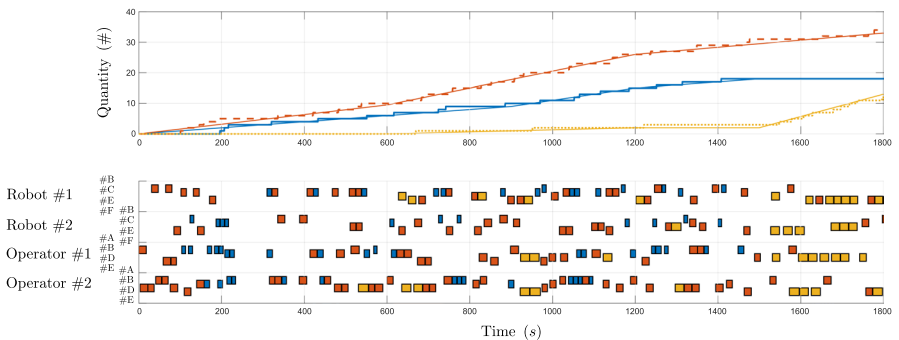
Fig. 9 Actual production for the 3 products (#1: solid blue, #2: dashed red, and #3: dotted yellow, respectively) compared to the corresponding reference (top), and Gantt chart of the operations (empty spaces correspond to either an idle time or a transfer between a station to another) (Color figure online)

6 Conclusions

A simulation-based robust scheduling and dispatching algorithm has been proposed in this paper to control high-mix production systems. The manufacturing system is regarded as a stochastic process and its simulation-based model is continuously updated using real-time analytics. Through simulations, the proposed algorithm is able to robustly select the optimal production strategy by dynamically allocating resources to jobs in the shop-floor. A simulated experiment consisting in an assembly layout with six stations and four resources has been introduced to validate the approach. The algorithm has been shown able to respond quickly to varying production requests, lot sizes, and variable takt-times, despite the variabilities in job processing times.

6.1 Managerial insights

Agile manufacturing is a broad term used to describe a production approach able to respond quickly to highly-varying customer demands or other factors such as changing lot sizes, variants, while still being able to maintain low production costs. In the case of high-mix and highly variable low-volumes production settings, collaborative robotics can surely help in providing the desired level of flexibility. On the other hand, the operation level requires fast evaluation of the production process in order to promptly dispatch and scheduler jobs to the available resources. The high number of variants prevents traditional systems based on static assignments and balancing to process orders in an optimal way. In this context, this research has developed an algorithm for optimal scheduling and dispatching rules that allow a production facility to leverage on human-robot collaboration and digitalisation to timely react to possible market volatilities. In particular, a digital twin of the manufacturing site is used to process many scenario analyses and decide which job assignment and sequencing policy are the most suited to be executed. The best policy is then selected in order to robustly minimise the deviation of future production with respect to the desired one, as retrieved from an ERP system. The main features of the developed method are:

- the possibility to schedule jobs in high-mix scenarios;
- the possibility to systematically handle uncertainties in the durations of jobs;
- the possibility to handle highly variable production requests thanks to the predictive approach.

6.2 Limitations and future directions

The methodology proposed in this work surely suffers from being completely agnostic about costs. It is worth noticing, in fact, that decisions regarding the next jobs to start and the corresponding dispatching is taken only based on a performance metric, and without considering its cost. This clearly sounds like a limitation in case alternative decisions might have significantly different costs, tough similar performance. This limitation should be handled introducing a more suitable definition of the cost

function, still relying on the same optimisation algorithm. For example, in case a reduced production request, moving around stations and waiting idle in a station are regarded as completely interchangeable alternatives, though the former requires a cost (in terms of energy), while the latter probably does not.

The main limitation of the method is probably due to the modelling part. The method requires a model of the production environment to be available. Though the method does not require a specific modelling paradigm, as the complexity of the plant increases, the modelling task becomes even more time consuming. The availability of engineering tools to speed-up this task can be clearly beneficial for the applicability of the method.

As for future research directions, when dealing with mobile platforms, a better and possibly automatically designed traffic management system will be a scope of possible further investigations. Moreover, the incorporation within the digital twin of non-nominal execution of jobs (including errors, re-work, need for programmed stops for maintenance, etc.) will be also worth of future studies. In fact, one of the assumptions of the method as described in the paper, is that every job can terminate in a finite time. While for manual activities this is surely true, automated tasks without a complete coverage in terms of error-recovery strategies cannot be considered by the method, at current stage.

Appendix

List of symbols adopted throughout the paper:

t	current time instant (s)
τ	generic (future) time instant (s)
$\mathbb{P}$	set of products, $\mathbb{P} = \left\{0, 1, \dots, N_{\mathbb{P}}\right\}$
$t_{i,p}$	timestamp of the i-th order of product $p \in \mathbb{P}$ (s)
$Q_{i,p}$	requested quantity of product $p \in \mathbb{P}$ within the i-th order (#)
$\Delta T_{i,p}$	takt-time of the i-th order of product $p \in \mathbb{P}$ (s)
Γ	duration of the prediction horizon (s)
$\theta_p(t)$	number of products of type p available at time t (#)
$\theta_p^0(\tau)$	desired number of products of type p available at time τ (#)
$\theta_{p,\pi}(\tau)$	predicted number of products of type p available at time instant τ according to policy π (#)

References

Bhosale K, Pawar P (2019) Material flow optimisation of production planning and scheduling problem in flexible manufacturing system by real coded genetic algorithm (rcga). Flex Serv Manuf J 31(2):381–423

Blazewicz J, Ecker KH, Pesch E, Schmidt G, Sterna M, Weglarz J (2019) Scheduling in flexible manufacturing systems. In *Handbook on Scheduling*, pages 671–711. Springer, Switzerland

Cardin O, Trentesaux D, Thomas A, Castagna P, Berger T, El-Haouzi HB (2017) Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. J Intell Manuf 28(7):1503–1517

Casalino A, Zanchettin AM, Piroddi L, Rocco P (2021) Optimal scheduling of human–robot collaborative assembly operations with time petri nets. IEEE Trans Autom Sci Eng 18(1):70–84. https://doi.org/10.1109/TASE.2019.2932150

Cassandras CG, Lafortune S (2009) Introduction to discrete event systems. Springer, Berlin

Cataldo A, Perizzato A, Scattolini R (2015) Production scheduling of parallel machines with model predictive control. Control Eng Pract 42:28–40

Dianatfar M, Latokartano J, Lanz M (2019) Task balancing between human and robot in mid-heavy assembly tasks. Procedia CIRP 81:157–161

Ding H, Schipper M, Matthias B (2014) Optimized task distribution for industrial assembly in mixed human-robot environments-case study on io module assembly. In *2014 IEEE international conference on automation science and engineering (CASE)*, pages 19–24

Dörmer J, Günther H-O, Gujjula R (2015) Master production scheduling and sequencing at mixed-model assembly lines in the automotive industry. Flex Serv Manuf J 27(1):1–29

Ferreira C, Figueira G, Amorim P (2018) Optimizing dispatching rules for stochastic job shop scheduling. In *International conference on hybrid intelligent systems*, pages 321–330. Springer

Glasserman P, Yao DD (1992) Some guidelines and guarantees for common random numbers. Manag Sci 38(6):884–908

Godinho Filho M, Barco CF, Neto RFT (2014) Using genetic algorithms to solve scheduling problems on flexible manufacturing systems (fms): a literature survey, classification and analysis. Flex Serv Manuf J 26(3):408–431

Gombolay M, Wilcox, R, Shah J (2013) In *Fast scheduling of multi-robot teams with temporospatial constraints conference*. Massachusetts Institute of Technology, Cambridge

Ivanov D, Dolgui A, Sokolov B (2016a) Robust dynamic schedule coordination control in the supply chain. Comput Ind Eng 94:18–31

Ivanov D, Dolgui A, Sokolov B, Werner F, Ivanova M (2016b) Schedule robustness analysis with the help of attainable sets in continuous flow problem under capacity disruptions. Int J Prod Res 54:3397–3413

Ivanov D, Sokolov B, Chen W, Dolgui A, Werner F, Potryasaev S (2021) A control approach to scheduling flexibly configurable jobs with dynamic structural-logical constraints. IISE Trans 53(1):21–38

Ivanov D, Sethi S, Dolgui A, Sokolov B (2018) A survey on control theory applications to operational systems, supply chain management, and industry 40. Annual Rev Control 46:134–147

Karaulova T, Andronnikov K, Mahmood K, Shevtshenko E (2019) Lean automation for low-volume manufacturing environment. *Annals of DAAAM and Proceedings*, 30

Krüger J, Lien TK, Verl A (2009) Cooperation of human and machines in assembly lines. CIRP Ann 58(2):628–646

Lin JT, Chiu C-C, Chang Y-H (2019) Simulation-based optimization approach for simultaneous scheduling of vehicles and machines with processing time uncertainty in fms. Flex Serv Manuf J 31(1):104–141

Li Q, Wang L, Xu J (2015) Production data analytics for production scheduling. In *2015 IEEE international conference on industrial engineering and engineering management (IEEM)*, pp. 1203–1207

Mateus JEC, Aghezzaf E-H, Claeys D, Limère V, Cottyn J (2018) Method for transition from manual assembly to human-robot collaborative assembly. IFAC-PapersOnLine 51(11):405–410

Messner M, Pauker F, Mauthner G, Frühwirth T, Mangler J (2019) Closed loop cycle time feedback to optimize high-mix/low-volume production planning. Procedia CIRP 81:689–694

Meyer H, Fuchs F, Thiel K (2009) *Manufacturing execution systems (MES): Optimal design, planning, and deployment*. McGraw Hill Professional

Moon YB, Phatak D (2005) Enhancing erp system's functionality with discrete event simulation. Ind Manag Data Syst 105(9):1206–1224

Morel G, Panetto H, Zaremba M, Mayer F (2003) Manufacturing enterprise control and management system engineering: paradigms and open issues. Annu Rev Control 27(2):199–209

Morioka M, Sakakibara S (2010) A new cell production assembly system with human-robot cooperation. CIRP Ann 59(1):9–12

Ostermeier FF (2019) The impact of human consideration, schedule types and product mix on scheduling objectives for unpaced mixed-model assembly lines. Int J Prod Res 58(14):4386–4405

Ouelhadj D, Petrovic S (2009) A survey of dynamic scheduling in manufacturing systems. J Sched 12(4):417

Pinedo M (2012) Scheduling, vol 29. Springer, Berlin

Ramadge PJ, Wonham WM (1987) Supervisory control of a class of discrete event processes. SIAM J Control Opt 25(1):206–230

Saif U, Guan Z, Zhang L, Zhang F, Wang B, Mirza J (2019) Multi-objective artificial bee colony algorithm for order oriented simultaneous sequencing and balancing of multi-mixed model assembly line. J Intell Manuf 30(3):1195–1220

Schlette C, Buch AG, Hagelskjær F, Iturrate I, Kraft D, Kramberger A, Lindvig AP, Mathiesen S, Petersen HG, Rasmussen MH et al (2020) Towards robot cell matrices for agile production-sdu robotics' assembly cell at the wrc 2018. Adv Robot 34(7–8):422–438

Shi Z, Gao S, Du J, Ma H Shi L (2019) Automatic design of dispatching rules for real-time optimization of complex production systems. In *2019 IEEE/SICE international symposium on system integration (SII)*, pages 55–60

Sprodowski T, Sagawa JK, Maluf AS, Freitag M, Pannek J (2020) A multi-product job shop scenario utilising model predictive control. Expert Syst Appl 162:113734

Tao F, Zhang H, Liu A, Nee AY (2018) Digital twin in industry: state-of-the-art. IEEE Trans Ind Inf 15(4):2405–2415

Tendeloo Y Van, Vangheluwe H (2018) Discrete event system specification modeling and simulation. In *Proceedings of the 2018 winter simulation conference*, Gothenburg, Sweden. pp 162–176. https://doi.org/10.1109/WSC.2018.8632372

Wang W, Hu Y, Xiao X, Guan Y (2019) Joint optimization of dynamic facility layout and production planning based on petri net. Procedia CIRP 81:1207–1212

Wu S-YD, Wysk RA (1989) An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing. Int J Prod Res 27(9):1603–1623

Yu C, Semeraro Q, Matta A (2018) A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. Comput Oper Res 100:211–229

Zeigler BP, Kim TG, Praehofer H (2000) Theory of modeling and simulation. Academic press, US

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Andrea Maria Zanchettin was born in Cremona (Italy) in 1983. He received his MSc in Computer Science Engineering and his PhD in Information Technology, both with honour, from Politecnico di Milano in 2008 and 2012, respectively. During Spring 2010, he spent a research stay at the Department of Automatic Control (Reglerteknik) at Lund University. From January 2012 until February 2014 he has been a temporary research assistant at the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB). From March 2014 until September 2019 he has been a fixed-term assistant professor at DEIB where he is now Associate Professor. His research interests are about mechatronic systems (robotics mainly) and

automatic control. Andrea Zanchettin has been member of the IEEE Robotics and Automation Society since 2009. Since 2017, he has been co-founder and co-chair of the IEEE RAS Technical Committee on Collaborative Automation for Flexible Manufacturing. Dr. Zanchettin is also chair of the Italian Chapter of the IEEE RAS (I-RAS). He is also co-founder and member of the Board of Directors of Smart Robots, a spin-off company of Politecnico di Milano.