Received 31 October 2023, accepted 7 December 2023, date of publication 12 December 2023,
date of current version 20 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3341504

RESEARCH ARTICLE

Dynamic MPC-Based Scheduling in a Smart Manufacturing System Problem

ALESSANDRO BOZZI[1,2], (Member, IEEE), SIMONE GRAFFIONE[1], (Member, IEEE), ROBERTO SACILE[1], (Member, IEEE), AND ENRICO ZERO[1], (Member, IEEE)

[1]Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS), University of Genoa, 16145 Genoa, Italy
[2]Laboratory System and Materials for Mechatronics (SYMME), Université Savoie Mont Blanc, 74940 Annecy, France

Corresponding author: Alessandro Bozzi (alessandro.bozzi@edu.unige.it)

ABSTRACT This paper introduces a dynamic scheduling algorithm designed to minimize makespan within a smart manufacturing system, accommodating delays in the production process. The proposed approach relies on Model Predictive Control (MPC) principles and adapts flow-shop scheduling theory to solve an open-shop scheduling problem. It aims to strike a balance between the ideal, delay-free solution and robustness in the case of processing time delays. By combining MPC theory with flow-shop scheduling, the algorithm offers a robust approach to open-shop scheduling problems, even with uncertain processing times. Iterated upon the arrival of each new job on the shop floor, the algorithm incorporates a control horizon to predict impending job arrivals and seamlessly integrates them into the scheduling process. Efficiency is examined through a comprehensive case study, where it is compared against a similar, offline scheduling algorithm. This novel method not only optimizes scheduling but also adapts to dynamic scenarios, reducing the computational demand and the information needed to optimize the production process, thus making it suitable for agile manufacturing environments. The results demonstrate the algorithm's efficacy in achieving competitive scheduling performance with nearly the same makespan as the offline algorithm, while accounting for uncertainties in processing times. A robustness analysis confirms the reliability of the proposed approach, showing an average improvement of 5% in makespan across different delay magnitudes.

INDEX TERMS Model predictive control, open shop problem, scheduling, smart manufacturing systems.

I. INTRODUCTION

Nowadays, manufacturing enterprises confront escalating and fierce market competition, driven by diverse customer demands and the rapid expansion of economic globalization. Hence, they have to extend their production mode into distributed environments and establish multiple factories in various geographical locations. Intelligent manufacturing systems, sometimes also referred to as smart manufacturing systems, are characterized by the integration of advanced technologies like artificial intelligence, Internet of Things (IoT), and connected machinery to monitor and optimize the production process [1]. A literature review on distributed scheduling problems in intelligent manufacturing systems is provided in [2]. Production scheduling stands as a fundamental aspect in manufacturing systems, aimed at optimizing crucial objectives including profitability, operational efficiency, and energy conservation. This optimization process revolves around critical factors such as determining the most efficient processing paths, machine allocation, and precise processing times.However, the intricate nature of large-scale, strongly-coupled constraints, combined with the demand for real-time solutions in specific scenarios, presents considerable challenges in effectively addressing manufacturing scheduling problems [3]. Competitive global market conditions have necessitated real-time agility and flexibility in manufacturing process industries. As the operational paradigm evolves, process scheduling decisions are increasingly focused on shorter time scales, and their interactions with the process control layer have gained significant importance [4].

In manufacturing systems, effectively managing product flow and resource allocation is of paramount importance to ensure satisfactory performance and minimize delays

The associate editor coordinating the review of this manuscript and approving it for publication was Jiefeng Hu.

and congestion in production processes. Various strategies can be employed to address these challenges, aiming to reduce idle and wasted time within the manufacturing process. This study presents a comparative analysis of offline and online strategies, illustrating distinct approaches and performance metrics within the context of an open-shop scheduling problem (OSSP). The OSSP, widely recognized for its extensive industrial applications, represents a pivotal concern within the realm of engineering [5]. Preventive maintenance plays a crucial role that must be considered in addressing these challenges, as seen in the context of an open-shop scheduling model that accounts for human errors and preventive maintenance considerations [6]. The proposed model takes into consideration conflicting objective functions including makespan, human error, and machine availability. To illustrate its applicability, a real case study is presented, demonstrating the practical application of the multi-objective mixed integer nonlinear programming model. A flexible job-shop scheduling problem with a random machine breakdown has been widely studied in [7]. This study simultaneously addresses two objectives: makespan and robustness, where the latter measures schedules' ability to withstand breakdowns. Similarly, this paper proposes a newly designed iterative algorithm to account for uncertainties in processing times and in order to provide a robust solution that minimizes makespan.

A predictive maintenance model to detect, monitor, and control emergent behavior by IoT sensors has been proposed to detect, monitor, and control emergent behavior, thereby mitigating sensor failures and system downtime [8]. Similarly, a comparison between offline and online algorithms is presented in the domain of electric vehicle charging within a single charging station [9]. Just as in our work, these algorithms are centralized and framed as mixed-integer programming problems. Furthermore, we introduce an online algorithm that iteratively engages the offline counterpart, effectively managing unforeseen future arrivals.

Diverse offline algorithms can be employed to tackle these NP problems, offering a broader solution space. The study in [5] illustrates how the genetic algorithm's selection phase significantly impacts solution quality and its proposed algorithm demonstrates superior performance in generating solutions when compared to other developed methods, highlighting advantages in terms of computational efficiency and objective values. In modern discrete flexible manufacturing systems, the occurrence of dynamic disturbances in real-time is a frequent challenge. Additionally, individual jobs may encompass various specialized operations due to technological requirements [10]. The proposed model is particularly well-suited for real-time scheduling within discrete flexible manufacturing facilities and designed to accommodate scenarios where jobs consist of multiple operations bound by the no-wait constraint, even with dynamic disturbances. To address stochasticity, typically associated with processing times, [11] developed a multiobjective scheduling model. This model seeks to optimize product quality while minimizing tardiness, which refers to the delay or lateness in completing scheduled tasks. Similarly to our work, [12] accomplishes a bi-objective optimization combining a Mixed Integer Linear Programming (MILP) model with reinforcement learning. MILP and Constrained Programming are commonly employed for job scheduling in various facilities. For instance, [13] provides a comparative analysis of these methodologies, elucidating their respective advantages and drawbacks. Similarly, [14] utilizes an MILP model and a memetic algorithm to optimize a bi-objective cost function, specifically minimizing total setup time and the number of late jobs. Furthermore, its applicability extends to diverse modern Industry 4.0 smart factories characterized by elevated complexity and stochasticity. Such environments demand reactive real-time scheduling methods to facilitate product individualization and accommodate product variety.

Online scheduling finds diverse applications across various case studies. A metaheuristic algorithm addressing real-time control problems in energy-efficient scheduling for flexible job shops, where rescheduling impacted operations and updating schedules is necessary, has been proposed [15]. The effectiveness of this approach is further illustrated through an energy-efficient scheduling case study for a flexible job shop. This case study demonstrates the ease and speed with which optimal schedules and accurate supervisory control instructions can be obtained. Machine learning plays a huge role in optimizing the schedules in flexible manufacturing system, as in [16]. Other works employ deep reinforcement learning, which revealed to be promising for short-term scheduling [17], [18], also in the need of adaptability within the shop [19], whereas long-term scheduling manages the product demand over a longer time horizon (i.e., weekly or monthly), taking into consideration also the costs of inventory and employment [20]. Additionally, in [21] Petri Nets and a heuristic based on artificial intelligence have been combined to solve scheduling within flexible manufacturing systems. In this work, a comparison between offline and online short-term scheduling is presented. The online algorithm adopts the Model Predictive Control (MPC) approach for scheduling forecasts. MPC is frequently employed in such problems, as seen in a proposal for a distributed model predictive control-based energy scheduling for islanded multi-microgrids [22]. This approach's performance is validated through a comparative analysis against existing models. Scheduling energy usage in smart homes poses a significant challenge due to the stochastic and unpredictable nature of many influencing factors. To tackle this, an MPC model has been suggested for modeling and managing uncertainties associated with smart home appliance scheduling [23].

The paper is organized as follows: Section II presents a comprehensive overview of the methodology employed to address the open shop scheduling problem, delineating both offline and online approaches. Section III introduces the case study, while Section IV examines the resultant outcomes in detail. Finally, Section V encapsulates the conclusions drawn from the study.

II. MATERAL AND METHODS

In an open-shop scheduling problem, the objective is to schedule J jobs on M machines to execute O_j operations each ($j = 1, \ldots, J$). Notably, in OSSP the number of operations for each job may differ and cycles (i.e. multiple passage of jobs on machines) may be present in their path, increasing the general complexity of finding a feasible and efficient solution. With the advancement of technology, in recent years the evolution in manufacturing has given birth to the flexibility of machines. In other words, certain machines are able to accommodate multiple operation types, thereby enabling distinct routing possibilities within the shop for each job. In this research, the open-shop scheduling problem is modeled as a flow-shop scheduling problem. This approach involves systematically enumerating all feasible paths that a job can follow within the shop, taking into account its specific production constraints. This enumeration results in a range of alternative paths, denoted as A_j, representing the various choices available for job j. These alternatives collectively form the set A, including all possible choices for each job, and it is represented as $A = A_1, A_2, \ldots, A_J$. This augmentation amplifies the system's adaptability and flexibility.

The interplay between jobs' routings can give rise to shared resources, representing the machines through which multiple jobs must pass during production. This leads to the formulation of sets of disjunctive connections, denoted as D, which ensures the orderly processing of one job at a time on shared machines. This dynamic interplay is illustrated in Fig. 1, where a generalized graph G visually captures the product flow through the shop. Alternatives are organized into rows by job, signifying potential sequencing on machines. It is worth noting that precisely one alternative must be selected for each job.

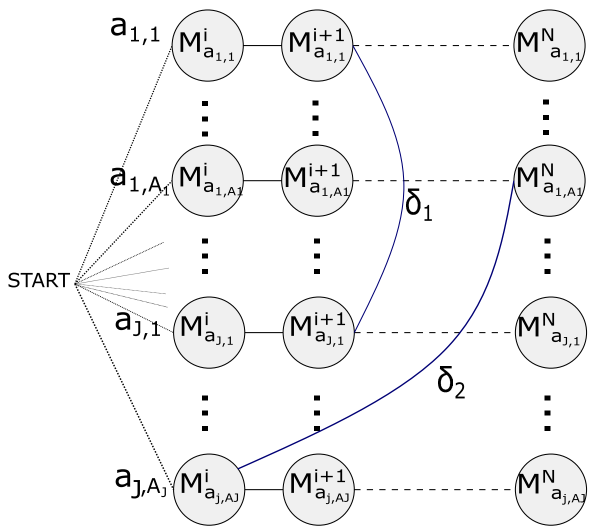


FIGURE 1. Flow-shop graph equivalent to the OSSP.

The diagram further highlights disjunctive connections, such as δ_1 and δ_2, demonstrating the interconnections between alternatives of different jobs on shared machines (e.g., $M^{i+1}_{a_{1,1}} = M^{i+1}_{a_{j,1}}$ and $M^{i+1}_{a_{j,A_J}} = M^{N}_{a_{1,A_1}}$).

The key challenge in solving this problem lies in identifying the optimal set of alternatives for each job while minimizing specific predefined criteria, such as the makespan. In other words, the goal is to select the most efficient combination of paths for each job that collectively results in the best overall solution in terms of makespan.

The novelty of this research lies in the implementation of a model predictive control (MPC) algorithm to dynamically solve an OSSP. Specifically, upon the arrival of a new job at the shop, the OSSP is tackled taking into account both the jobs currently present on the shop floor and those that, under the known planning, fall within the selected prediction horizon for the MPC.

To enhance the realism of the problem, the arrival of jobs, though anticipated during the planning phase, is modeled using a zero-mean Gaussian noise, introducing an element of unpredictability.

The re-scheduling algorithm is triggered with each actual arrival of a new job onto the shop floor. This mechanism ensures that the system remains adaptive and responsive to the dynamic changes introduced by real-world arrivals.

A. PROBLEM FORMULATION

In order to systematically address the optimization problem through mixed integer linear programming (MILP), it is essential to establish a clear framework by introducing the relevant sets, variables, and constants that define the key parameters governing the system.

Therefore, the sets are:

- J: number of jobs;
- M: number of machines;
- D: number of disjunctive connections on shared machines;
- A: number of alternatives;
- O_j: number of operations for each job j.

Meanwhile, the decision variables of the problem are:

- $s_{j,m} \in \mathbb{R}$, $[J \times M]$: the start time of job j on machine m;
- $c_{j,m} \in \mathbb{R}$, $[J \times M]$: the completion time of job j on machine m;
- $\delta_d \in \{0, 1\}$, $[D \times 1]$: variable for disjunctive connections between shared resources;
- $\gamma_{j,a_j} \in \{0, 1\}$, $[A \times 1]$: variable for modeling possible job's choices;
- $C \in \mathbb{R}$, $[1 \times 1]$: completion time of the last job that completes the production process.

The constants of the problem are:

- P, $[J \times M]$: matrix of ideal processing times, where $p_{j,m}$ corresponds to the processing time of job j on machine m. Processing times can be affected by disturbances that cause delays in production;

- $\mathbb{M}$: big-M, used to activate and deactivate pairs of constraints;
- R_p, $[J \times 1]$: vector of jobs' planned release time. It is subject to Gaussian disturbances that may anticipate or delay their arrival time at the shop floor. Consequently, R_r represents the vector of the real release time of jobs.

The associated mixed integer programming problem minimizes the completion time of jobs:

$$\min_{\underline{s},\underline{c}} C \tag{1}$$

subject to constraints (2a) to (2g):

$$s_{j,m(j,o)} \geq c_{j,m(j,o-1)} - (1 - \gamma_{j,a_j})\mathbb{M} \quad \forall j = 1, .., J \quad \forall o = 2, \dots, O_j \tag{2a}$$

$$c_{j,m} = s_{j,m} + p_{j,m} \sum_{a_j=1}^{A_j} \gamma_{j,a_j} \quad \forall j = 1, .., J \quad m = 1, \dots, M \text{ and } a_j = 1, ..A_j \tag{2b}$$

$$s_{j_1,m} \geq c_{j_2,m} - \delta_d \mathbb{M} \qquad \forall j_1, j_2 \in D \tag{2c}$$

$$s_{j_2,m} \geq c_{j_1,m} - (1 - \delta_d)\mathbb{M} \qquad \forall j_1, j_2 \in D \tag{2d}$$

$$s_{j,m(j,1)} \geq R^{real}_{0_j} \qquad \forall j = 1, .., J \tag{2e}$$

$$C \geq c_{j,m(j,O_j)} \qquad \forall j = 1, .., J \tag{2f}$$

$$\sum_{a_j=1}^{A_j} \gamma_{j,a_j} = 1 \qquad \forall j = 1, .., J \tag{2g}$$

Specifically, constraint (2a) establishes the relationship between the start and completion time of consecutive machines in the sequence of job j, conditioned to the selected alternative. The notation $m(j, o)$ serves to denote the machine that processes job j at its o^{th} operation in the sequence. Constraint (2b) ensures the adherence to the correct processing time within each machine. The disjunctive constraints are represented in (2c) and (2d), where the δ_d variables guarantee the activation of only one of the two constraints for each shared machine. Furthermore, (2e) ensures compliance with the job release times in the shop, while (2f) determines the completion time of the last processed job. Lastly, (2g) addresses the γ variables, ensuring that for each job, a single alternative is chosen, thereby avoiding multiple paths for the same job.

In summary, the optimization problem aims to determine the optimal routing to minimize job waiting times and completion time, while adhering to resource availability, machine occupancy, and job processing flow constraints.

B. GUARANTEEING EFFICIENCY WITH STOCHASTIC PROCESSING TIMES

The objective of this work is also to provide a robust solution that remains effective even when processing times are disrupted. To ensure a realistic behavior of the system, two key assumptions have been taken into account: each machine can, at most, double its processing time, and the total disturbance is bound by an upper limit denoted as Ω. To achieve this, an iterative algorithm has been designed to strike the optimal balance between a solution unaffected by noise and a resilient solution capable of handling worst-case scenarios, in which disturbances on machines maximize the delay in production. Thus, the possible delay of job j on machine m is represented with a new decision variable $w_{j,m}$. The optimization cost function then translates into a minimax problem: The former minimization problem, which focused on reducing the makespan, has evolved into a minimization of the makespan while accounting for the maximum delays, representing the most adverse machine delay scenarios:

$$\min_{\underline{s},\underline{c}} \max_{\underline{\omega}} C \tag{3}$$

Constraints (2a)-(2g) remain valid, except for constraint (2b) which needs to be extended to accommodate delays and be independent of γ:

$$c_{j,m} = s_{j,m} + p_{j,m} + (p_{j,m}\omega_{j,m}) \quad \forall j = 1, .., J; m = 1, \dots, M \text{ and } a_j = 1, ..A_j \tag{4}$$

Then, additional constraints are needed to realistically model delays:

$$\sum_{j=1}^{J} \sum_{i=1}^{A_j} \gamma_{i,j} \sum_{m=1}^{M} \omega_{j,m} = \Omega \tag{5a}$$

$$0 \leq \omega_{j,m} \leq 1 \quad \forall j = 1, \dots J \quad m = 1, \dots, M \tag{5b}$$

In (5a), the optimization problem ensures that only the $\omega_{j,m}$ values associated with the alternatives selected in the previous minimization problem are considered in the computation of the delay configuration. Furthermore, constraint (5b) enforces an upper limit on the magnitude of each disruption, ensuring that, in the worst-case scenario, each machine is allowed to at most double its processing time, and not beyond.

1) FINDING THE BEST TRADE-OFF SOLUTION

In order to find a solution capable of keeping competitive performance both in the case of deterministic and stochastic processing times, with known magnitude Ω, the following iterative approach has been employed:

- **Step 1**: Solve the problem outlined in Section II-A, which represents the minimum achievable completion time for the given scenario in the ideal, deterministic case
- **Step 2**: Set the recently determined γ variables as parameters for the max sub-problem, while keeping the δ variables to be determined through the optimization problem. This involves imposing the path for each job while using the sequencing on machines and disturbance on processing time, namely δ and ω, as decision variables. A solution for ω represents the worst-case scenario in the given routing, while still allowing flexibility in sequencing on machines
- **Step 3**: Revisit the problem detailed in Section II-A. In this step, set $p_{j,m} = p_{j,m} + p_{j,m}\omega_{j,m} \quad \forall j = 1, \dots, J; m = 1, \dots, M$ and solve the problem anew.

Subsequently, compare the new job path solution (i.e., only the γ variables) with the one obtained in Step 1:

- If the two solutions coincide, it indicates that the job path represents the best trade-off between optimality and robustness when confronted with potential delays of the given magnitude
- Should the new solution differ from its predecessor, it suggests that the delay-free solution is no longer optimal provided conditions (5a) and (5b) hold. In such an instance, to derive an appropriate solution for both scenarios, the γ variables of the novel solution are once again set as parameters, and Step 2 is reiterated.

This approach enables the identification of a job path that performs well across a range of Ω values, thus accommodating potential delays, even if guaranteed convergence is not assured. Consequently, the algorithm is intentionally halted after a predetermined number of iterations, with the selection of the solution minimizing the completion time in the deterministic scenario.

C. MPC-BASED SCHEDULING FORMULATION

The problem discussed in Sec. II-A and Sec. II-B pertains to the offline scheduling of jobs through machines. In ideal situations, possessing comprehensive insight into the entire production process becomes crucial to ensure the accuracy of the solution. However, there might be instances where this information is unavailable, or the only accessible data is the planned timing of job releases over time. As a result, the aforementioned problem needs an expansion to tackle the scheduling challenge in real-time fashion, with each new product arrival. In order to accomplish this, it is necessary to keep track of the progress of jobs that have undergone operations within the shop. Thus, at each event denoted as t, marking the arrival of a new product, the γ representing paths that are no longer feasible, due to the job having already completed a portion of the route between machines, rendering its passage through other machines impossible, are removed from the job's potential alternatives. Analogously, the machines currently assigned to jobs are retained until their completion time, to avoid their assignment to other jobs when solving the optimization problem.

$$s^{t}_{j,m} = s^{t-1}_{j,m} \tag{5c}$$

$$c^{t}_{j,m} = c^{t-1}_{j,m} \tag{5d}$$

where t represents the t^{th} event (i.e. arrival of a job) on the shop floor.

These constraints enable the execution of the scheduling algorithm described in the previous sections for every arrival of a product. Additionally, a prediction horizon can be incorporated to facilitate the scheduling of jobs currently present on the shop floor and those planned to arrive shortly (i.e. within the prediction window). Consequently, the set J, which represents the set of jobs considered in the optimization problem, is initially a subset of the complete job pool. As the final job arrives on the shop floor, this set J encompasses the entire spectrum of jobs.

Thus, for each new event (i.e. arrival of one or more jobs):

$$\begin{aligned} &\forall \mathrm{p} \in \mathcal{P}, \forall \mathrm{s} \in \mathcal{S} \left(\mathrm{s}_{\gamma} \cap \mathrm{p}_{\gamma} = \emptyset, \mathcal{P}_{\gamma} \subseteq \Gamma \right) \\ &\quad \mathit{with} \quad \# (\mathcal{S} \cup \mathcal{P}) = J \end{aligned} \tag{6}$$

where

- $\mathcal{P}$ contains all the jobs within the prediction horizon;
- p is a job belonging to set $\mathcal{P}$;
- $\mathcal{S}$ contains all the jobs that have been already scheduled or are being processed;
- s is a job belonging to set $\mathcal{S}$;
- Γ is the set of all alternatives;
- Pedix γ refers to the chosen alternative for the job of the corresponding set.

This translates into a dynamic, reactive controller capable of scheduling jobs based on the system's current state while forecasting the imminent arrival of new products. Fig. 2 graphically represents the implemented algorithm. This formulation has been solved with the approach described in the following sections.

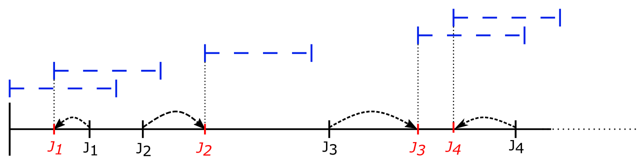


FIGURE 2. Graphical representation of the MPC-based scheduling. Black dashes represent the planned arrival time of jobs, and red dashes their real arrival time. Blue lines indicate the prediction horizon of the algorithm, triggered at each new arrival of a job and needed to figure out which jobs are considered in the scheduling problem.

III. CASE STUDY

To assess the effectiveness of the proposed approach, a theoretical case study is devised, involving $M = 6$ machines and $J = 6$ jobs. The complete range of feasible paths is outlined in Table 1, in which each machine is represented as a number for sake of notation. The table highlights that all jobs are required to commence from M_1, which serves as the loading unit of the SMS. Furthermore, it is notable that jobs have to perform from 4 to 5 operations, and certain alternatives may share segments of the initial path, thereby enhancing the MPC-based scheduling's ability to dynamically select the optimal route depending on the real-time release of other jobs. The flexibility of machines is expressed by Table 2, while the planned and real release time of jobs is listed in Table 3. The planned release time for a job can deviate by a maximum of 3 units of time, either ahead of schedule or delayed. In contrast, the MPC's prediction horizon spans 2 units and inherently considers the scheduled release time due to the unknown actual release time. Moreover, jobs are assumed to keep the scheduled order, avoiding possible swaps arising from concurrent delays and advancements of consecutive jobs.

TABLE 1. Alternative paths for each job.

Job	Alternative	O_1	O_2	O_3	O_4	O_5
J_1	γ_1	1	2	3	4	5
	γ_2	1	3	5	6	
J_2	γ_3	1	2	3	4	6
	γ_4	1	2	4	5	
J_3	γ_5	1	2	5	6	
	γ_6	1	2	3	5	
	γ_7	1	3	4	6	
J_4	γ_8	1	2	6	5	3
J_5	γ_9	1	4	6	5	3
	γ_{10}	1	2	4	5	6
J_6	γ_{11}	1	3	4	5	
	γ_{12}	1	2	3	4	6
	γ_{13}	1	3	4	6	

TABLE 2. Processing times.

	M_1	M_2	M_3	M_4	M_5	M_6
J_1	9	5	7	10	4	12
J_2	4	7	3	7	1	10
J_3	5	7	6	3	10	1
J_4	4	3	10	6	4	5
J_5	2	4	7	3	5	2
J_6	1	6	5	3	6	8

TABLE 3. Jobs' planned and real release time.

	R_0^{plan}	R_0^{real}
J_1	0	0
J_2	2	0
J_3	4	2
J_4	7	4
J_5	10	7
J_6	12	14

The prediction horizon is a critical parameter that should be tailored to the specific characteristics of the production order. It determines how far into the future the scheduling algorithm looks when making decisions. On the one hand, a longer prediction horizon allows the scheduler to consider more jobs at once, providing a more comprehensive view of the production landscape. On the other hand, it results in increased computational complexity, as more jobs are taken into account simultaneously and the optimization problem becomes more complex and time-consuming to solve. Consequently, scheduling with a longer prediction horizon may lead to a slight increase in computational time and could exhibit performance characteristics similar to those of offline scheduling.

IV. RESULTS

To solve the MILP problem presented in this study, MATLAB has been employed, coupled with the Optimization toolbox, a powerful optimization tool designed for mathematical programming and linear programming tasks. The performance of the proposed approach is compared to an offline scheduling algorithm, deeply explained in [24]. An offline scheduling approach incurs higher computational demands and necessitates complete, deterministic knowledge of the system for its operation. Despite this, it has the potential to yield superior makespan performance.

Conversely, an MPC-based scheduling approach can achieve performance levels on par with offline scheduling, while also demonstrating the capability to adapt to unanticipated job arrivals or machinery malfunctions. Machine breakdowns, although relevant, have not been addressed in this study. Furthermore, this method leverages the system's current and near-future knowledge (i.e. within the prediction horizon) to formulate suboptimal job schedules.

It is worth noting that while offline scheduling excels in optimal performance but requires extensive upfront information, MPC-based scheduling provides a balanced approach that accounts for real-world dynamics and unpredictabilities, making it a compelling choice for agile manufacturing environments.

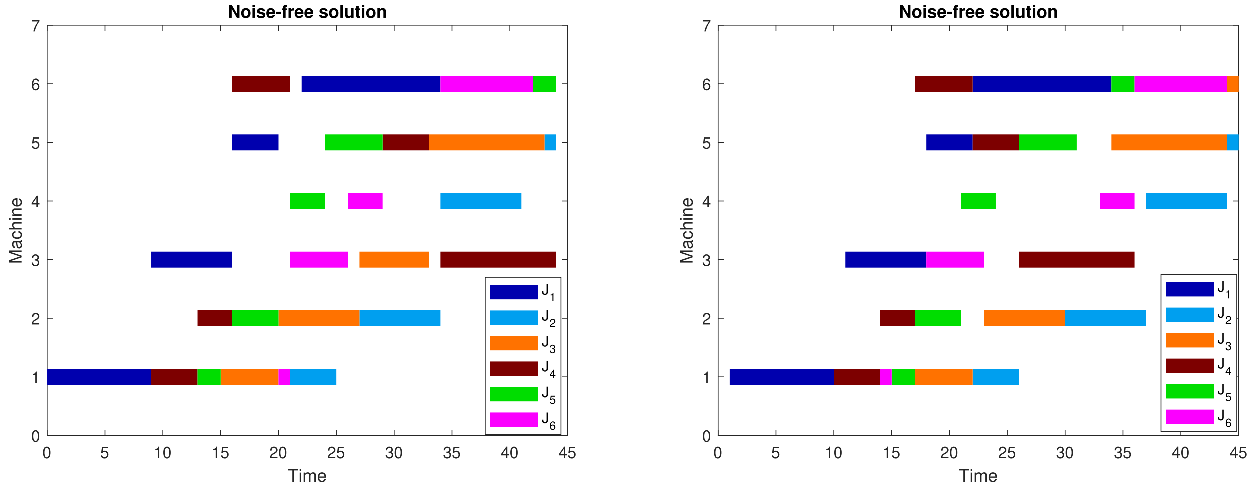
Fig. 3 illustrates the optimal solutions in the absence of disturbances in processing times. It can be observed that the completion time is nearly identical: 44 using the offline technique and 45 with the MPC-based approach. The key distinction lies in the fact that the latter does not necessitate deterministic knowledge of the entire system but rather dynamically adapts to the introduction of new jobs within the shop floor. This reasoning is highlighted in Table 4, which

TABLE 4. MPC dynamic choice of the routing in the noise-free scenario at each arrival of new jobs.

Jobs in the shop	Chosen alternatives
J_1	γ_2
J_1, J_2	γ_2, γ_4
J_1, J_2, J_3	$\gamma_2, \gamma_4, \gamma_6$
J_1, J_2, J_3, J_4	$\gamma_2, \gamma_4, \gamma_6, \gamma_8$
J_1, J_2, J_3, J_4, J_5	$\gamma_2, \gamma_4, \gamma_5, \gamma_8, \gamma_{10}$
$J_1, J_2, J_3, J_4, J_5, J_6$	$\gamma_2, \gamma_4, \gamma_5, \gamma_8, \gamma_{10}, \gamma_{13}$

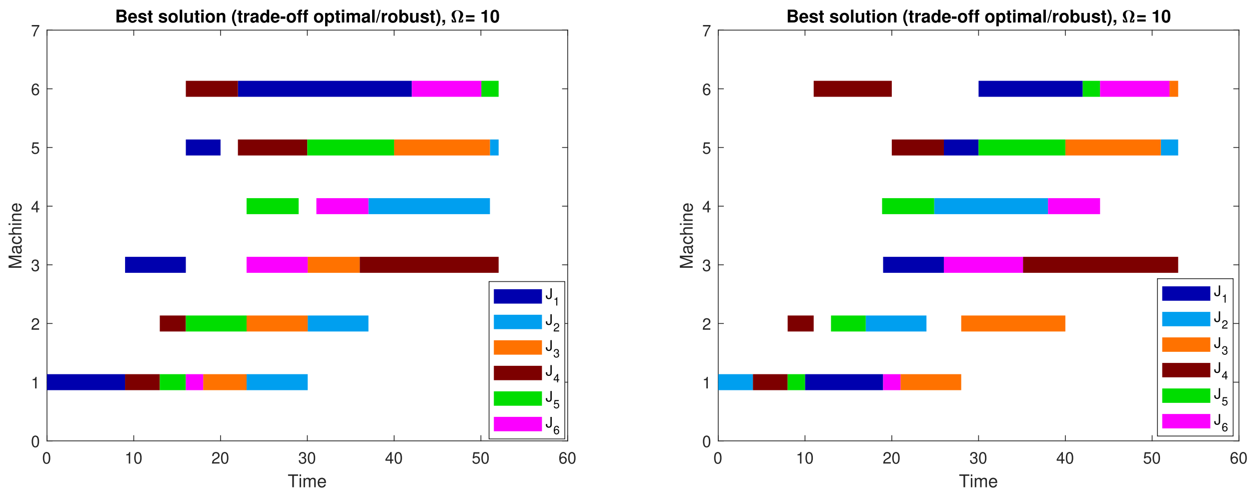
refers to the noise-free scenario and points out the chosen path at each arrival of a new job in the shop. The adaptability of the MPC-based scheduling becomes evident in the context of J_3. Specifically, it is initially scheduled according to the routing specified by γ_6. However, upon the subsequent arrival of J_5, J_3 is dynamically rescheduled using the routing strategy defined by γ_5. This dynamic adjustment remains feasible because J_3 has not yet been allocated to a machine that would render altering its routing unviable, and it ensures the maintenance of competitive performance levels.

In order to shorten the computational time needed to solve the optimization problem, the alternatives listed in Table 1 are automatically eliminated once a job follows a divergent path incompatible with the continuation of the previously chosen alternative. Similarly, Fig.4 depicts a comparison between solutions found with the iterative approach of Sec. II-B when processing times are perturbed, with a magnitude of $\Omega = 10$, while considering the worst-case distribution across machinery of these disturbances. In this scenario, similarly as in the case of deterministic processing time, the completion time remains nearly identical, despite the presence of disturbances.



((a)) **Offline,** $\Omega = 0$ ((b)) **MPC-based,** $\Omega = 0$

FIGURE 3. Comparison between offline (a) and MPC-based (b) scheduling solutions in the case of deterministic processing times. Both solutions are comparable in terms of completion time.



((a)) **Offline,** $\Omega = 10$ ((b)) **MPC-based,** $\Omega = 10$

FIGURE 4. Comparison between offline (a) and MPC-based (b) scheduling solutions in the presence of delayed processing times. Completion time is 52 in the ideal offline scenario, representing the minimum achievable value, and 53 in the dynamic scheduling, thus maintaining competitive performance.

Furthermore, from a computational standpoint, the MPC-based algorithm offers substantial benefits. Although more computations are involved in managing data structures and implementing the controller, it is important to highlight that the initial schedulings only involve a subset of the total jobs, resulting in faster optimization. As the number of jobs increases, on the other hand, much of the routing within the shopfloor for already present jobs has already been determined, thus minimizing its involvement in the optimization problem. Consequently, the pool of potential alternatives to be analyzed is significantly reduced. For these reasons, the implementation of the MPC-based algorithm results in an enhancement of computational performance. In the examined case study, it has been observed a significant 30% reduction in execution time, and it is reasonable to assume that a larger number of jobs corresponds to a more pronounced reduction in computational time, due to the algorithm's operational approach. Indeed, at each iteration, the MPC-based algorithm exclusively schedules jobs currently inside the shop floor as well as those anticipated to imminently arrive (i.e. within the prediction horizon). This selective scheduling enables the algorithm to effectively manage a limited number of jobs with each invocation, as opposed to offline scheduling.

To further substantiate the validity of the approach, Fig. 5 presents a robustness analysis conducted through 1000 simulations for the MPC-based scheduling, for each value of Ω. Processing times were subjected to random,

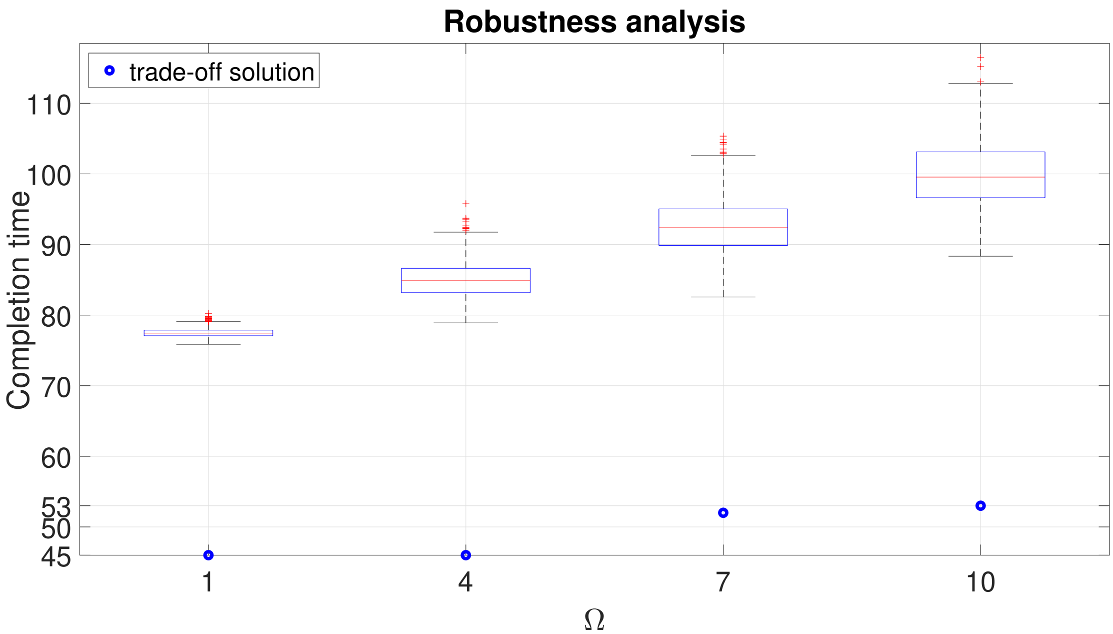


FIGURE 5. Robustness analysis of 1000 simulations for each magnitude of noise. Solutions present the same routing, different feasible sequencing on machines, and random noises generated according to a uniform probability distribution. Blue points represent the solutions obtained with the proposed algorithm in the case of worst delays.

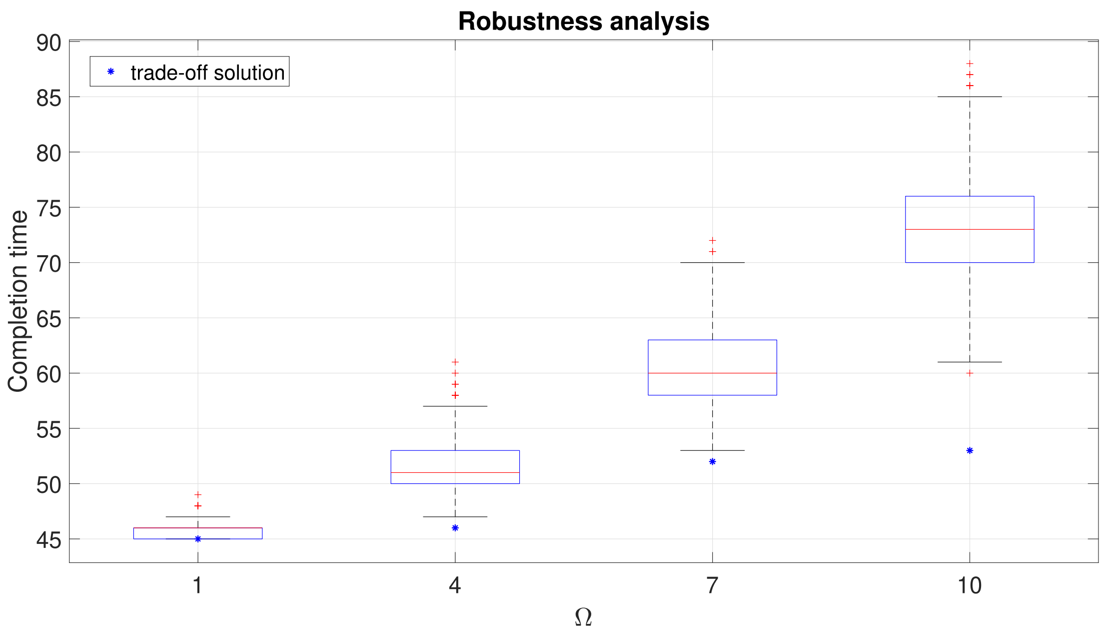


FIGURE 6. Robustness analysis of 1000 simulations for each magnitude of noise. Solutions present the same routing, same sequencing on machines, and random noises generated according to a uniform probability distribution. Blue points represent the solutions obtained with the proposed algorithm in the case of worst delays.

uniformly distributed perturbations across the machinery, and a feasible solution with the same routing (i.e., identical γ values) was successfully determined. Similarly, as shown in Fig. 6, the solutions share identical routing and machine sequencing, differing only in the randomly generated uniformly distributed disturbances that affect processing times. The figures demonstrate that, across a wide range of scenarios, the approach consistently delivers superior performance. It strikes an optimal balance between the ideal solution without delays and a robust solution when the magnitude of delays is known for the given OSSP.

V. CONCLUSION

This research builds upon the foundation laid in [24] and extends it by incorporating MPC-based scheduling to address an OSSP. Incorporating a prediction horizon enables dynamic adaptation to new job arrivals, the events that trigger the algorithm to solve the optimization problem again, while eliminating the requirement for an exhaustive understanding of the entire planned system and real release times. Simulations have showcased comparable performance, even in scenarios characterized by highly disturbed processing times, that can be attributed to the algorithm's capacity to reschedule jobs on machines, thereby maintaining effectiveness. This study presents a solution that accommodates both the ideal case, where processing times remain undisturbed, and scenarios involving disrupted processing times. The proposed solution offers adaptability suitable for a diverse range of potential scenarios, achieved through the recalibration of optimal scheduling each time a new job enters the system. This iterative approach ensures the solution remains responsive and flexible, capable of dynamically addressing changes as they occur.

Furthermore, an evaluation of computational performance, although slightly dependent on code skills and hardware specifics, has been conducted. The results reveal expedited computations, further reinforcing the proposition of employing the algorithm for real-time scheduling within agile manufacturing systems.

Future research will involve the practical validation of the approach in real-world scenarios, with a comprehensive performance analysis to identify potential time constraints for the given set of problems. Moreover, future endeavors will encompass the management of machinery faults and subsequent job rescheduling along the production chain. This extension of the approach to handle unforeseen disruptions will allow for a more comprehensive evaluation of its effectiveness within real-world manufacturing environments.

ACKNOWLEDGMENT

The authors acknowledge the use of ChatGPT by OpenAI to refine the academic language and accuracy of this work.

They have made the code accessible for future developments and the addition of new features through the following GitHub repository: https://github.com/Bozzi96/MPC_scheduling.git.

REFERENCES

[1] B. Wang, F. Tao, X. Fang, C. Liu, Y. Liu, and T. Freiheit, "Smart manufacturing and intelligent manufacturing: A comparative review," *Engineering*, vol. 7, no. 6, pp. 738–757, Jun. 2021.

[2] Y. Fu, Y. Hou, Z. Wang, X. Wu, K. Gao, and L. Wang, "Distributed scheduling problems in intelligent manufacturing systems," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 625–645, Oct. 2021.

[3] L. Wang, Z. Pan, and J. Wang, "A review of reinforcement learning based intelligent optimization for manufacturing scheduling," *Complex Syst. Model. Simul.*, vol. 1, no. 4, pp. 257–270, Dec. 2021.

[4] J. M. Simkoff and M. Baldea, "Production scheduling and linear MPC: Complete integration via complementarity conditions," *Comput. Chem. Eng.*, vol. 125, pp. 287–305, Jun. 2019.

[5] A. A. Rahmani Hosseinabadi, J. Vahidi, B. Saemi, A. K. Sangaiah, and M. Elhoseny, "Extended genetic algorithm for solving open-shop scheduling problem," *Soft Comput.*, vol. 23, no. 13, pp. 5099–5116, Jul. 2019.

[6] M. Sheikhalishahi, N. Eskandari, A. Mashayekhi, and A. Azadeh, "Multi-objective open shop scheduling by considering human error and preventive maintenance," *Appl. Math. Model.*, vol. 67, pp. 573–587, Mar. 2019.

[7] Y. Yang, M. Huang, Z. Y. Wang, and Q. B. Zhu, "Robust scheduling based on extreme learning machine for bi-objective flexible job-shop problems with machine breakdowns," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113545.

[8] R. Sacile, M. Sallak, and E. Zero, "Detection and classification of failures as an emergent behavior in a machinery system modelled as a system of systems," in *Proc. 18th Annu. Syst. Syst. Eng. Conf. (SoSe)*, Jun. 2023, pp. 1–7.

[9] A.-M. Koufakis, E. S. Rigas, N. Bassiliades, and S. D. Ramchurn, "Offline and online electric vehicle charging scheduling with V2V energy transfer," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2128–2138, May 2020.

[10] S. Luo, L. Zhang, and Y. Fan, "Real-time scheduling for dynamic partial-no-wait multiobjective flexible job shop by deep reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3020–3038, Oct. 2022.

[11] Y. Fu, H. Wang, J. Wang, and X. Pu, "Multiobjective modeling and optimization for scheduling a stochastic hybrid flow shop with maximizing processing quality and minimizing total tardiness," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4696–4707, Sep. 2021.

[12] Y. Du, J.-Q. Li, X.-L. Chen, P.-Y. Duan, and Q.-K. Pan, "Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem," *IEEE Trans. Emerg. Topics Comput. Intell.*, pp. 1–15, 2022.

[13] S. Fatemi-Anaraki, R. Tavakkoli-Moghaddam, M. Foumani, and B. Vahedi-Nouri, "Scheduling of multi-robot job shop systems in dynamic environments: Mixed-integer linear programming and constraint programming approaches," *Omega*, vol. 115, Feb. 2023, Art. no. 102770.

[14] Z. Zhao, S. Liu, M. Zhou, and A. Abusorrah, "Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 6, pp. 1199–1209, Jun. 2021.

[15] S. Tian, T. Wang, L. Zhang, and X. Wu, "An energy-efficient scheduling approach for flexible job shop problem in an Internet of Manufacturing Things environment," *IEEE Access*, vol. 7, pp. 62695–62704, 2019.

[16] M. H. Abidi, H. Alkhalefah, M. K. Mohammed, U. Umer, and J. E. A. Qudeiri, "Optimal scheduling of flexible manufacturing system using improved lion-based hybrid machine learning approach," *IEEE Access*, vol. 8, pp. 96088–96114, 2020.

[17] C. D. Hubbs, C. Li, N. V. Sahinidis, I. E. Grossmann, and J. M. Wassick, "A deep reinforcement learning approach for chemical production scheduling," *Comput. Chem. Eng.*, vol. 141, Oct. 2020, Art. no. 106982.

[18] X. Wang, L. Zhang, Y. Liu, and C. Zhao, "Logistics-involved task scheduling in cloud manufacturing with offline deep reinforcement learning," *J. Ind. Inf. Integr.*, vol. 34, Aug. 2023, Art. no. 100471.

[19] L. Wang, X. Hu, Y. Wang, S. Xu, S. Ma, K. Yang, Z. Liu, and W. Wang, "Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107969.

[20] F. Hanssmann and S. W. Hess, "A linear programming approach to production and employment scheduling," *Manage. Sci.*, vol. 1, no. 1, pp. 46–51, Jan. 1960.

[21] A. R. Moro, H. Yu, and G. Kelleher, "Hybrid heuristic search for the scheduling of flexible manufacturing systems using Petri nets," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 240–245, Apr. 2002.

[22] A. Nawaz, J. Wu, J. Ye, Y. Dong, and C. Long, "Distributed MPC-based energy scheduling for islanded multi-microgrid considering battery degradation and cyclic life deterioration," *Appl. Energy*, vol. 329, Jan. 2023, Art. no. 120168.

[23] B. Liang, W. Liu, L. Sun, Z. He, and B. Hou, "Economic MPC-based smart home scheduling with comprehensive load types, real-time tariffs, and intermittent DERs," *IEEE Access*, vol. 8, pp. 194373–194383, 2020.

[24] A. Bozzi, S. Graffione, J. Jiménez, R. Sacile, and E. Zero, "Reliability evaluation of emergent behaviour in a flexible manufacturing problem," in *Proc. 18th Annu. Syst. Syst. Eng. Conf. (SoSe)*, Jun. 2023, pp. 1–7.

ALESSANDRO BOZZI (Member, IEEE) received the joint master's degree in complex and interacting systems from the University of Genoa, Italy, and the University of Compiègne, France, in 2020. He is currently pursuing the Ph.D. degree in systems engineering with the University of Genoa and Université Savoie Mont Blanc. His research interests include autonomous vehicle platooning, system of systems control, and scheduling within smart manufacturing systems.

SIMONE GRAFFIONE (Member, IEEE) received the joint degree in the system of systems and the master's degree in computer engineering from the University of Technology of Compiegne, France, in 2019. He is currently pursuing the Ph.D. degree in system engineering with the University of Genoa. His research interest includes system engineering and its application in the automotive sector.

ROBERTO SACILE (Member, IEEE) received the master's and Ph.D. degrees in electronic engineering from the University of Genoa, Italy, in 1990 and 1994, respectively. He has been with the University of Genoa, since 2000, where he is currently a Researcher and a Professor of automation and systems engineering. His research interests include systems engineering and its application to the transportation, logistics, and energy sectors.

ENRICO ZERO (Member, IEEE) received the Ph.D. degree in systems engineering approaches to safety in transport systems (computer science and system engineering) from the University of Genoa, in 2022. His research interests include the interactions of automotive data and physiological signals during the road transport of dangerous goods.