Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

WORKLOAD CONTROL IN HIGH-MIX-LOW-VOLUME FACTORIES THROUGH THE USE OF A MULTI-AGENT SYSTEM

Jeroen B.H.C. Didden
Quang-Vinh Dang
Ivo J.B.F. Adan

Department of Industrial Engineering
& Innovation Sciences
Eindhoven University of Technology
Eindhoven, The Netherlands

ABSTRACT

Order release in High-Mix-Low-Volume machine environments is often difficult to control due to the high variety of these shops. This paper, therefore, proposes an extension to a Multi-Agent System to control order release. Intelligence is introduced to the agent that is responsible for order release to autonomously learn which jobs to release into the shop through the use of sequencing rules, depending on the current environment. The objective is to minimize the mean weighted tardiness of all jobs. Computational results show that the proposed sequencing rules outperform other more common dispatching rules in terms of mean weighted tardiness. Further analysis of the results also reveals that a more accurate prediction of the lead time of jobs can be made, which is one of the main interests of practitioners in High-Mix-Low-Volume environments.

1 INTRODUCTION

A new emerging paradigm within Industry 4.0 is the fast-changing customer portfolio (Mourtzis 2016). Customers desire both short lead times and high customisability, which enhances the complexity of current manufacturing systems (Bertrand and Van Ooijen 2002). Information regarding jobs is therefore often only known at the last moment, creating a real-time planning and scheduling problem. The various types of products to be produced increases drastically, while the demand for each individual type is extremely low. In addition, products become more complex, causing an increase in both specialized equipment and processing steps during production. This creates the concept of High-Mix-Low-Volume (HMLV) production (Didden et al. 2021).

One way for better control of manufacturing lead times is through order release (Thürer et al. 2016). Instead of immediately releasing jobs into the system upon entry, jobs are stored in a pre-shop pool. From this pre-shop pool, jobs are then released periodically or when certain conditions are met to control performance measures (Thürer et al. 2015). The overall goal of workload control is to create smaller and more predictable flow/lead times, which can, in turn, help to reduce the overall tardiness of jobs within the system. Multiple methods have been proposed in the literature to control the flow of jobs within the shop, such as Workload Control (Thürer et al. 2012; Bertrand and Van De Wakker 2002), Constant Work-In-Process (Bertolini et al. 2020; Prakash and Chin 2014; Thürer et al. 2017), Kanban (Thürer et al. 2016), and Drum-Buffer-Rope (Darlington et al. 2015; Thürer and Stevenson 2018). Each of these methods differs in the specific way they handle order release. However, the main difference is whether the total workload of the shop is limited (e.g., Workload Control) or whether the number of jobs on the shop floor is limited (e.g., CONWIP).

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

However, one of the difficulties faced when implementing order release methods in HMLV production environments is correctly controlling the flow of jobs that enter the shop (Gómez Paredes et al. 2020). The large variation in total processing time of each job can strictly limit the number of jobs present within a workstation, causing starvation or blocking of other workcenters. In addition, especially in complex machine shops, setup times become highly important (Thürer et al. 2012). Releasing many different job types into the shop at once can drastically increase the machines' total setup time, further increasing the overall tardiness. Especially when job information is only known at the last minute, correctly controlling the flow becomes more important. Even though some research has looked at which jobs should be released into the shop (i.e., the pre-shop sequencing decisions) (Thürer et al. 2015; Neuner and Haeussler 2021), research regarding the effect of these sequencing rules in HMLV environments is still lacking.

One method to solve these planning and scheduling problems in HMLV factories is through Multi-Agent Systems (MAS) (Zhou et al. 2019; Sahin et al. 2017; Maoudj et al. 2019; Wang et al. 2016; Ebufegha and Li 2021). A substantial amount of research has been done on the integration of MAS in machine shops, as well as order release methods, yet the combination of the two is lacking in the literature. Weng et al. (2008) are one of the only studies to integrate an order release method within a MAS. Here, the authors use an Earliest Due Date sequencing rule and workload norms to release jobs onto the shop floor where results show that the decentralization of the shop floor through the MAS also helps to improve order release methods, especially during the sequencing of the pre-shop pool. More information becomes available due to the cooperation of agents on the shop floor, allowing for more complex sequencing rules to be created. Further, as agents within a MAS can perceive their environment, they can also learn from it, allowing for an improved method of order release.

In this paper, we propose an extension to a previously proposed MAS by Didden et al. (2021). The objective is twofold: (1) enhance the previously proposed MAS by including a more complex Job Release Agent and (2) dynamically learn the pre-shop sequencing pool of the Job Release Agent in different settings. The remainder of the paper is structured as follows. Section 2 provides an overview of the machine shop scheduling problem. Next, the enhanced MAS is presented in Section 3, while the learning algorithm to develop new sequencing rules for the pre-shop pool, shop floor characteristic, and simulation settings are given in Section 4. Experimental results to show the effectiveness of the new rule is given in Section 5. Finally, conclusions and interesting future research directions are discussed in Section 6.

2 PROBLEM DESCRIPTION

The problem considered is an online job shop scheduling problem within an HMLV environment. The environment consists of a number of workcenters W and a number of machines M. Each workcenter w contains a number of machines M^w, where $M^w \subseteq M$. A job j has an arrival time r_j to the shop according to a Poisson process, with a mean inter-arrival time of λ. Each job j is of a certain type $\tau \in \mathscr{T}$, where $\tau(j)$ denotes the type τ of job j. A job type τ defines the priority ω^τ and the ordered set of operations $O^\tau = \{(\tau,0), 0 = 1,\ldots,n_\tau)\}$ of job j, where $(\tau,o) \rightarrow (\tau,o+1)$ and n_τ is the number of operations for a job type τ. Each operation has a deterministic processing time p_o^τ, where $o \in O^\tau$. A job type τ also defines its routing $\pi^\tau = \{w_o^\tau \,|\, w_o^\tau \in W, o \in O^\tau\}$ throughout the shop (i.e., the order in which it has to visit specific workcenters), where w_o^τ is the workcenter of the *oth* operation of job type τ. Lastly, upon arrival at the shop, a due date $\bar{d}_j$ is assigned to each job j according to $\bar{d}_j = r_j + k\sum_{o\in O^{\tau(j)}} p_o^\tau$, where k is the due date tightness factor.

The goal of the system is to determine, in real time, in which order jobs to release to the shop floor. Additionally, the objective is to minimize the mean weighted tardiness of all jobs, i.e., $\min \frac{\sum_{j\in J} T_j}{|J|}$, where $T_j = \max(\omega^{\tau(j)}(f_j - \bar{d}_j))$ is the tardiness of a job j, J is the set of jobs that have finished processing, and f_j is the finishing time of a job j. In addition, the following assumptions are also made:

- A machine can only process a single operation at a time.
- Preemption is not allowed.

- Job information is only known upon arrival of a job to the shop.
- An operation of a job can only be processed after the previous operation of the same job has finished processing (i.e., precedence relations).
- Setup times $s_{j_1,j_2} > 0$ when jobs j_1 and j_2 are of different types, otherwise $s_{j_1,j_2} = 0$.
- All jobs are accepted upon arrival, and material is always available.

3 MULTI AGENT SYSTEM DESCRIPTION

This section enhances the MAS proposed in Didden, Dang, and Adan (2021) with a new job release agent. This MAS consists of three different layers, each with a number of agents responsible for the flow of orders and jobs throughout that level. The main focus of this study is on the relation between the agents within the Global Control Unit (GCU) and Shop Floor Planning (SFP) layers, see Figure 1.

Within the remaining layer, the Enterprise Resource Planning (ERP) layer, only two agents exist; (1) the ERP agent that communicates the information of accepted order towards the GCU and (2) the Database Agent that stores information regarding the planning and scheduling decisions of all agents to be later used during simulation optimization. Given that it is assumed that all orders are accepted, and the Database Agent is only responsible for the storage of data, no further elaboration is required on these agents.

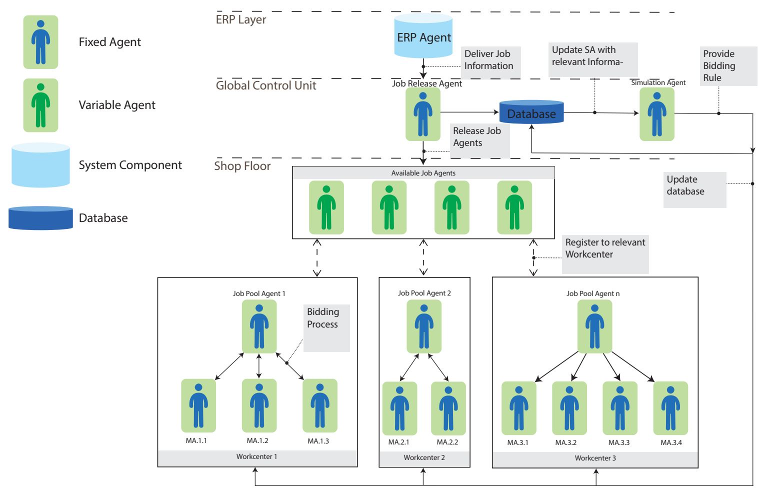

Figure 1: Overview of the proposed MAS.

3.1 Global Control Unit and Shop Floor Planning

Within the GCU and SFP layers of the system, multiple agents collaborate and corporate to optimize the flow of jobs throughout the entire production process. The GCU contains the Job Release Agent (JRA) that decides when to release jobs to the shop floor and which jobs to release. The SFP layer, on the other hand, contains three different agents: (1) Job Pool Agents (JPA), (2) Machine Agents (MA), and (3) Job Agents (JA). The latter only contains information regarding the job it corresponds to, e.g., current operations, due date, and processing times. It is therefore not described in detail in the following sections.

3.1.1 Job Release Agent

Compared to standard releasing strategies used for scheduling in the literature, i.e., immediate release, the Job Release Agent does not release orders immediately upon arrival. Jobs can be released based on the current number of jobs in the shop, e.g., through CONWIP or Kanban methods, or by considering the WIP in the shop, e.g., Periodic or Continuous order release. However, the LUMS-COR (Lancaster University Management School Correcter Order Release) method as proposed in Thürer et al. (2015) is implemented within the agent, as this has previously proved to be the best method, due to its combination between periodic and continuous release. This method works as follows:

1. Sort all jobs $j \in \mathscr{J}^R$ in ascending order of preference
2. The first job in the sorted list is considered for release first
3. If the corrected workload $\frac{p_o^{\tau(j)}}{o}$ at workcenter w, where the o^{th} operation of job j has to be processed, and the current workload $\mathscr{W}^w$, fits within the workload norm N^w of workcenter w, i.e., $\frac{p_o^{\tau(j)}}{o} + \mathscr{W}^w \leq N^w \; \forall o \in O_j$, then job j is released onto the shop floor. The workload levels are then updated for each workcenter job j has to visit, i.e., $\mathscr{W}^w = \frac{p_o^{\tau(j)}}{o} + \mathscr{W}^w \; \forall o \in O_j$.
4. Return to step 2 if there are still jobs left in the pool to be considered for release. Otherwise, stop the release procedure.

Note that the load contribution of an operation is given in terms of its corrected load. Hence, it is important to distinguish between the direct load $\mathscr{W}_D^w$ and indirect load $\mathscr{W}_I^w$ of a workcenter, where $\mathscr{W}^w = \mathscr{W}_D^w + \mathscr{W}_I^w$. The direct load consists of the WIP that currently has to be processed within the workcenter. In contrast, the indirect load consists of the WIP that is expected to still arrive at the workcenter. This difference defines the second method of the LUMS-COR. When the direct workload of a workcenter w drops to 0, then an extra order release event is triggered. Given the current pool of jobs, the first sequenced job from which the first operation falls within the workcenters with zero direct workload is immediately released to the shop floor. Here, it is disregarded whether the other operations of the job fall within the workload norms of the workcenters.

3.1.2 Job Pool Agent and Machine Agent

In the SFP layer, the JPA and MA are responsible for dispatching and sequencing jobs currently in the system. Each workcenter contains a single JPA and multiple MAs, corresponding to the number of machines within that workcenter. The task of the JPA is to dispatch jobs to the MA within its workcenter. First, the JPA creates a pool $\mathscr{J}^p$ consisting of all operations that currently have to be processed within the workcenter. Note that JAs register themselves to the JPA of the current operations that have to be processed within that job (note that the JA can only register to a single work center for every operation). Next, the JPA initiates a Contract Net (CN) Protocol (Poslad 2007). This is a special behavior of agents which represents an auction. Within this CN, the JPA first sends out a Call For Proposal (CFP) to the MAs, where a bid is requested from the MAs. Within the CFP, information regarding the jobs in the pool is sent. The MAs then each determine a bid for all jobs within the CFP. The MA then replies to the CFP with a *reply* message stating which job it wants and the respective bid for the job. The MA, therefore, only bids on a single job, the job for which it formulated the highest bid. After all the MAs have responded, the JPA decides which machine responded with a bid for each job in its pool and which of those machines submitted the highest bid. After the winner with the highest bid has been determined, the JPA sends and *Accept-Propose* message to the machines that receive a job and a *Refuse-Propose* message to the machines that do not receive a job.

In addition to participating in the CN, MAs are also responsible for sequencing the jobs that they receive during the CN. This is carried out through the use of a preference rule, similar to those used during the bidding procedure described in the previous paragraphs. Every time an operation has finished processing on the machine that an MA represents, the MA sorts the current jobs in its queue, labeled as $\mathscr{J}^m$, with m the index of the MA, in descending order of preference. It then selects the first job in this list to process next. On occasions, it may occur that no jobs are currently in its queue, and in that case, it waits until it receives a job during a CN.

4 CREATION OF SEQUENCING RULES FOR JOB RELEASE AGENT

This section proposes a model consisting of an estimation of distribution algorithm (EDA) (Kim et al. 2021) and a discrete-event simulation (DES) to optimize the preference rules used by the JRA and MAs during planning and scheduling. The EDA is used to create and update the preference rules' parameters, while the DES is used to test the generated priority rules on a simulated shop floor.

4.1 Estimation of Distribution Algorithm

The presented JRA and MA agents require preference rules to either evaluate jobs or sort the jobs in their current pool. A common method of doing this is through the use of (composite) dispatching rules. These rules are formed of either a single rule (e.g., Shorted Processing Time (SPT) or Earliest Due Date (EDD)) or a combination of two or multiple rules (e.g., SPT + EDD). However, it is known that not one dispatching rules work well in all scenarios. In addition, specifying a unique preference rule for each agent in the system can help to improve the performance of the system further. Therefore, it is chosen to learn a unique preference rule for both the JRA and MAs through a linear composite dispatching rule. This linear rule consists of a weighted sum of various attributes related to jobs that are considered during release, bidding or sequencing, and certain parameters related to other agents in the system (such as workload) that can influence the shop. Define $\mathscr{A}^x$ as the set of attributes considered during evaluation by agent x, the preference index value for a job j, defined as b_j is given as: $b_j = \sum_{i=1}^{\mathscr{A}^x} w_{ix} \cdot a_{ix}$, where w_{ix} is the weight parameter for attribute i of agent x and a_{ix} is the attribute i for agent x. Note that given the decentralized structure of the MAS, some agents only have access to their own local information, such that global information cannot be used as an attribute. In order to learn the weight parameters for all agents, a similar approach as presented in Kim et al. (2021) is used, i.e., consisting of an EDA. The EDA generates a number of individuals N each iteration, which it used to evaluate and generate the weight parameters. In short, the following steps are taken:

1. Fit a normal uni-variate gaussian to each weight parameter, with a mean $\mu_{ix} \in \mu$ and standard deviation $\sigma_{ix} \in \sigma$.
2. For each agent x, attribute $i \in \mathscr{A}^x$ and individual $n \in N$, sample a value $\eta_{ix} \in \mathscr{N}(0, e^{\sigma_{ix}})$. Create two weight parameters w^{n+} and w^{n-}. Store the collection of weight parameters for one individual in w^{n+} and w^{n-}, with $w_{ix}^{n\pm} \leftarrow \mu_{ix} \pm \sigma_{ix}$.
3. Evaluate the performance of all w^{n+} and w^{n-}. Use the simulation results to update μ and σ, with the help of a gradient calculation and an optimizer (e.g., Stochastic Gradient Descent or Adam, see Ruder (2016)).
4. Repeat from step 1 until σ converges to 0 or for a predefined number of steps.

4.2 ATTRIBUTE DESIGN FOR JOB RELEASE AGENT

Specific attributes have to be chosen for the JRA, as shown in Table 1. The choice of these attributes is based on some well-known dispatching rules, as previously described by Holthaus and Rajendran (1997) and Thürer et al. (2015). These attributes relate either to the jobs in the pool or to the workcenters. The Processing Time (PT) attribute relates to the increase of the direct workload of the first workcenter visited by a job. The Arrival Time (AT) states when a job entered the system (i.e., when the JA has been created). The Planned Release Date (PRD) indicates when a job should be released to the shop floor, i.e., subtracting the current mean flow time of jobs in the shop from the due date of the job ($\bar{d}_j - \bar{f}$), with $\bar{f}$ being the mean flowtime. The Remaining Due Date (RDue) states the time until the due date of a job, and Total Work Content (TWC) is the sum of all processing times for all operations of a job (i.e., the total workload contribution of a job to the shop). The Critical Ratio (CR) is denoted as the ratio between RDue and TWC. The Job Priority (JP) is the priority of a job that is closely related to the chosen objective function. Lastly, the Direct Workload of First Workcenter (DWFW) gives the total direct workload a job will encounter for its first operation upon release, while Number of Operation (NO) states how many operations a job has. Lastly, to increase the learning of the weight parameters, all attributes are normalized in the range $[0,1]$.

Table 1: Attributes considered by Job Release Agent

Attribute ID	Name	Abbreviation	Normalization Range	Equation
1	Processing Time	PT	[0, 35]	$p_1^{\tau(j)}$
2	Arrival Time	AT	[-400, 0]	$r_j - t$
3	Planned Release Date	PRD	[-250, 50]	$\bar{d}_j - \bar{f}$
4	Remaining Due Date	RDue	[-400, 1232]	$\bar{d}_j - t$
5	Total Work Content	TWC	[0, 154]	$\sum_{o \in O^{\tau(j)}} p_o^{\tau(j)}$
6	Critical Ratio	CR	[-10, 14.5]	$(\bar{d}_j - t)/\sum_{i \in O^{\tau(j)}} p_o^{\tau(j)}$
7	Job Priority	JP	[1, 10]	$\omega_j^{\tau(j)}$
8	Direct Workload of First Workcenter	DWF	[0, 500]	$\mathscr{W}_D^w \| w = \pi_1^\tau$
9	Number of Operations	NO	[0, 6]	$n_{\tau(j)}$

4.3 SHOP FLOOR CHARACTERISTICS

For the shop configuration, an HMLV environment was recreated based on relevant information gathered from industry partners. The considered shop floor consists of 6 different workcenters, 21 machines, and 20 job types. For each job type, a uniform distribution is used to generate the number of operations, processing times for each operation, job priority, setup times, and demand, as seen in Table 2, as this corresponds well to real-life data. The operations of each job type are assigned to a random work center, taking into account that a job can only visit a work center once. Next, according to the total workload encountered in a work center, machines are divided between workcenters. The mean inter-arrival time λ of jobs is calculated according to Equation (1):

$$\lambda = \frac{\bar{p} \cdot |\bar{O}|}{|M| \cdot \mu}, \tag{1}$$

where μ is the mean utilization of the shop, $\bar{p}$ the mean processing time of all operations, and $|\bar{O}|$ the mean number of operations per job. The actual inter-arrival time is sampled from an exponential distribution during simulation, replicating Poisson arrivals of jobs. The due date tightness level k of jobs is chosen from a uniform distribution upon job entry to the shop.

Table 2: Shop floor settings.

Parameter	Symbol	Value
No. Job Types	$\mathscr{T}$	6
No. of Machines	M	21
No. of Workcenters	W	6
Processing Times	p_{jo}^τ	$U \sim (10,35)$
Mean Inter Arrival Time	λ	6.3
Utilization	μ	90%
Setup Time	$\{s_{j_1,j_2} > 0 \| \tau(j_1) \neq \tau(j_2)\}$	$U \sim (0, 0.2 * \max(p_{jo}^\tau))$
No. of Operations per job type	O_j^τ	$U \sim (3,6)$
Job Priority	ω^τ	$U \sim (1,10)$
Job Demand	δ^τ	$U \sim (1,100)$
Due Date Tightness	k	$U \sim (4,8)$

4.4 SIMULATION SETTINGS

In order to assess the performance of the simulation runs, the shop is first loaded according to a poisson process, with 1500 jobs to create a steady state situation (i.e., the mean flow time of jobs on the shop floor is fairly consistent). Next, jobs are continuously loaded into the shop until jobs 1500 to 3500 have all finished processing. The shop floor performance is based on these 2000 jobs, where the data of the first 1500 loaded jobs is disregarded. Additionally, a termination criterion is used in order to terminate bad runs early. This termination criterion is based on the number of jobs in the JRA, $\mathscr{J}^R_{\max}$, and a time limit $t_{\max}$. If the termination criterion is reached during simulation, the simulation is immediately stopped, and the objective function of the current run is set to $\sum_{j\in J_c} \max(\omega_j(f_j - \bar{d}_j), 0) + C - J_c$. Here, J_c is the set of completed jobs at the end of termination, and C is a sufficiently high penalty value to give a high value to incomplete simulations, as provided by (Branke et al. 2015).

Other settings used during the experiments are given in Table 3. The learning parameters α_μ and α_σ used in the ADAM optimizer are set to have an exponential decay, i.e., $\alpha = \alpha_0 \cdot \exp(-\frac{p-1}{\Delta})$, where Δ is the learning decay rate, α_0 the initial learning rate, and p the current population number (i.e., the current iteration of the EDA). The initial values α^0_μ, α^0_σ, and σ_0 are set according to the values stated in Kim et al. (2021), where σ_0 is the initial standard deviation. The learning parameters, β_1 and β_2, related to the ADAM optimizer, are set as stated in Kingma and Ba (2015). Lastly, the JRA tries to release jobs to the shop floor every 17.0 time units, labelled $\mathscr{J}_t$, which is close to the mean processing time of all operations.

Table 3: Experimental settings.

Setting	Symbol	Value
Population Size	P	$N_{attr} + 3$
Learning Decay Rate	α	1000
Optimizer	-	ADAM
Initial Learning Rate Mean	α^μ_0	0.1
Initial Learning Rate Standard Deviation	α^σ_0	0.025
Adam Parameter 1	β_1	0.9
Adam Parameter 2	β_2	0.999
Number of Jobs	N_{jobs}	2000
Initial Standard Deviation	σ_0	$log 0.3$
Initial weight Parameters JRA	w_{ax}	0
Range of Workload Norms	N^w	$[500, 550, 600, 650, 700]$
JRA Periodic Release	$\mathscr{J}_t$	17.0
Max job in JRA	$\mathscr{J}^R_{\max}$	100
Maximum time	$t_{\max} t$	100000

5 RESULTS AND DISCUSSION

To learn the weight parameters of the JRA, the EDA and DES are both integrated into Python, where Simpy is used to model the job shop as a DES. All experiments were conducted on a PC with an AMD Ryzen 9 5950X 16-Core Processor@4.9 GHz with 32 GB of RAM. The mean run time of the EDA is approximately an hour; the run time of a single DES after learning the weights is approximately 1.0 seconds. Note that the attributes for dispatching and sequencing of the MAs are designed similarly to those of the JRA. However, further details are omitted as the experiments focus on the influence of the JRA on the system.

Furthermore, to show the effectiveness of learning the pre-shop pool preference rule for the JRA, the results are compared to commonly used dispatching rules as proposed in the literature as presented in Table 4 (see e.g., Thürer et al. (2015)). Note that Γ is a sufficiently large number to prioritize urgent jobs

Table 4: Pre-shop sequencing rules as proposed in literature

Abbreviation	Name	Equations
FCFS	First Come First Serve	$\min_{j\in\mathscr{J}^R} r_j$
EDD	Earliest Due Date	$\min_{j\in\mathscr{J}^R} \bar{d}_j$
PRD	Planned Release Date	$\min_{j\in\mathscr{J}^R} \bar{d}_j - \bar{f}$
CR	Critical Ratio	$\min_{j\in\mathscr{J}^R} (\bar{d}_j - t)/ \sum_{o\in O^{\tau(j)}} p_o^{\tau(j)}$
MODCS	Modified Capacity Slack	$\begin{cases} \frac{\sum_{o\in O^{\tau(j)}} \left(\frac{\frac{p_o^{\tau(j)}}{o}}{N^W - \mathscr{W}^W}\right)}{O^{\tau(j)}} & \text{if } \bar{d}_j - \bar{f} \leq t \\ \bar{d}_j - \bar{f} + \Gamma & \text{if } \bar{d}_j - \bar{f} > t \end{cases}$

(i.e., exceeded their planned release times). Performance is compared across 4 different measures, namely (1) Mean Weighted Tardiness, (2) Max Weighted Tardiness, (3) Mean Flow Time, and (4) Percentage of Tady Jobs. The weight parameters for the JRA are learned over five different workload norms, ranging from 500 to 700. The results of these experiments are presented in Figure 2 and Table 5, where the mean performance is shown over 50 different simulation runs.

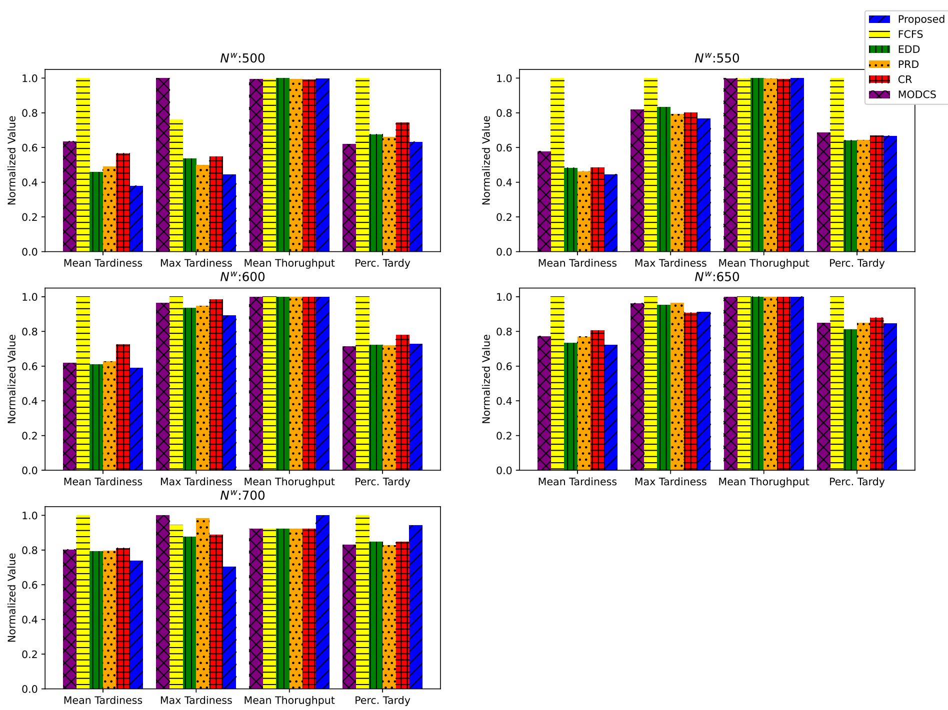

Figure 2: Comparison of the proposed method for sequencing jobs in the pre-shop pool of the JRA and other proposed methods.

What can initially be seen in Figure 2 is that the proposed method outperforms the other methods mainly in terms of mean and max weighted tardiness, with the differences being most notable at workload

Table 5: Percentage improvement of the proposed method compared to the next best sequencing rule.

$\mathcal{N}^W$	Mean Tardiness	Max Tardiness	Mean Thorughput	Perc. Tardy
500	20.61	12.44	-0.54	-2.19
550	3.74	3.36	-0.51	-3.91
600	3.34	5.05	-0.17	-2.20
650	1.56	-0.50	-0.16	-3.88
700	7.49	24.69	-7.84	-12.19

norms of 500 and 700. One can explain these differences through the objective function, where the mean weighted tardiness is aimed to be minimized. It especially becomes clear when looking at the other two performance measures. The mean throughput times are fairly close, with no large differences being seen except at a workload norm of 700. Here, the proposed method shows a higher flow time than the other methods. Often, a reduction in flow time will also result in a reduction in tardiness. However, in this case we are dependent on when the JRA releases a job to the shop floor. If a job is released to the shop floor late, then a low flow time does not necessarily decrease the tardiness. If a job is released early, on the other hand, then even a high flow time can still cause the job to be on time.

Moreover, another interesting observation can be made when looking at the combination of mean and maximum weighted tardiness and percentage tardy. For the proposed method, the percentage of tardy jobs is most of the times higher than in some of the other methods. However, both the mean and maximum weighted tardiness is always lower. This can be due to two reasons. First, the proposed method can give preference to jobs with a high priority, i.e., a high value of ω_j^τ, causing more jobs with a low priority to be late but directly decreasing the mean and maximum weighted tardiness. An extra test has been done by replacing the EDD rule with a Weighted-EDD rule. However, this caused an increase in both mean and max weighted tardiness, meaning that the JRA is less sensitive to the priority of jobs and more to their actual due date. A reasoning for this is that priority is more important during dispatching and sequencing, instead of order release. The second reason is that the distribution of tardy jobs is smaller for the proposed method, as shown in Figure 3. It is shifted more to the left when compared to using planned release dates as a sequencing rule (i.e., the rule resulting in the least percentage of tardy jobs). Therefore, even though the proposed method causes more jobs to be tardy, the overall tardiness of each job is still lower. This helps to estimate the flow time better, as the distribution of tardy jobs is much smaller (i.e., the standard deviation with respect to the proposed method is also smaller).

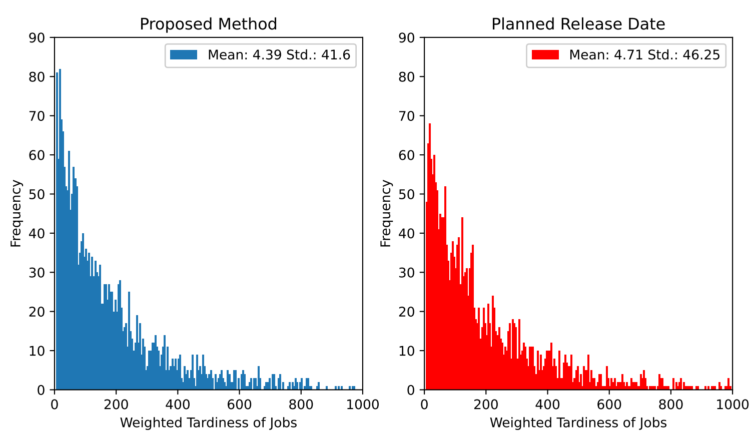

Figure 3: Comparison of the distribution of tardy jobs for the proposed rule and the PRD rule.

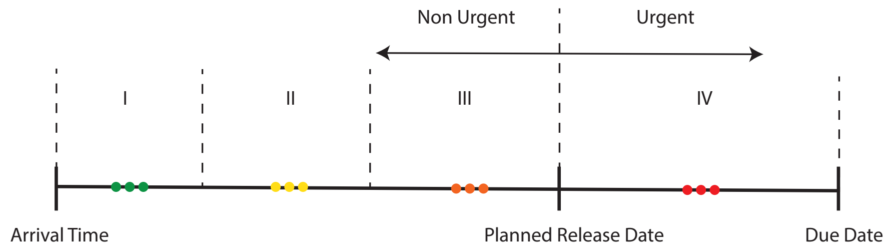

Figure 4: Different periods of job release considered by the JRA. Here, green means jobs are not urgent (i.e., far away from their PRD, while red jobs are urgent (i.e., passed their PRD).

Table 6: Weights of JRA for $N^W = 700$.

Attribute	PT	AT	PRD	RDue	TWC	CR	JP	DWF	NO
Weight	1.571	2.724	2.524	-2.593	0.314	2.919	-0.352	-0.191	-1.545

Looking at Table 6, where the learned weights for a workload norm of 700 are presented, some interesting examinations can be made. Note that for the JRA, the job with the lowest index value of b_j is considered for release first. Therefore, a positive value for the JRA means that the JRA prefers a low value for the attribute (i.e., close to 0 considering the normalization range), while a negative value means that the JRA prefers a high value for the attribute (i.e., close to one for the normalization range). E.g., the positive value in front of AT shows that jobs that have been waiting the longest since arrival are considered for release first.

Table 6 shows that the attributes AT, PRD, RDue, and CR dominate the other attributes, as they have either a very high or low associated weight. This means that preference is given to jobs that have been waiting the longest since arrival, close to their planned release date and have a low critical ratio (i.e., a low ratio between their remaining due date and total work content). These jobs are those in areas (III) and (IV) in Figure 4. The weight relevant to the remaining due date is the most interesting, as it gives preference to jobs with a high remaining time until the due date of their first operation. On the other hand, the high weight associated to CR can cancel out the negative weight of RDue, as CR also contains the remaining due date. However, this is only when jobs start to get a bit urgent, see area (II) in Figure 4. When a large amount of jobs in the pre-shop pool become tardy (i.e., when their RDue becomes small), then AT, PRD, and CR dominate the RDue attribute. In this case, jobs are chosen that are closer to their due date and planned release date.

Further, if all jobs in the pre-shop pool have enough slack (i.e., a high RDue, long time to their PRD, and close to their AT), see area (I) in Figure 4, then a job with the lowest processing time of its first operations, low TWC, and a high NO is chosen (i.e., the contribution of the attributes RT, PRD, RDue, and CR is low, as none of the jobs are tardy). More noticeable is the negative weight assigned to DWF. This causes a job to be chosen that has a high workload in the workcenter of its first operation, in cases the attributes related to the due dates and arrival time between jobs are similar. The JRA therefore aims to dispatch short jobs as quickly as possible, in order to fill up the system. With this, the increase in flow time can be explained. The JRA aims to fill up workstations to their norm to overcome starvation or help decrease the amount of setup times (i.e., when more jobs are available within a workcenter, there is a higher probability that two jobs of the same type or two jobs with a low setup time between them are present). This, however, will also directly cause higher flow times in the shop, as more jobs are present. However, as discussed in the previous paragraph, jobs are often released into the shop a long time before they reach their due dates, meaning that the increase in flow time can still cause the job to be on time, thus not directly causing a decrease in tardiness.

We derive two insights. First, when the machine shop is less busy, the system tends to release jobs close to AT, with low workload and a high number of operations, aiming to fill up workcenters as much as possible to efficiently utilize the machines without sacrificing tardiness. This is in contrast to what is often proposed in literature, to first send high workload jobs to the shop floor (see e.g., Thürer et al. (2015)). Secondly, when jobs are becoming urgent, and the machine shop is getting busier, the system tends to hold onto non-urgent jobs and, instead, it switches to choose jobs close to their due date and planned release date.

6 CONCLUSION

This paper proposes a new method to design adaptive sequencing rules for releasing jobs into a machine shop. Recent studies often focused on designing a single rule to sequence jobs. However, in the case of high-mix-low-volume production, a single rule often does not work well due to the dynamic environment where these shops are located. To overcome this, this paper discusses an extension to a proposed MAS, as well as the integration of an EDA to create preference rules dynamically. These preference rules are integrated into an agent, the JRA, that further uses the LUMS-CORS order release method to release jobs. Communication, cooperation, and collaboration between agents within the shop help increase the effectiveness of this method, as more information becomes available to different agents.

Results show that the proposed method of dynamically creating preference rules within the MAS outperforms other common preference rules in the literature, mainly in terms of mean and maximum weighted tardiness. In terms of percentage of jobs tardy and flow time, the proposed method gets outperformed for some settings of the maximum norm level. However, the combination of all four measures causes both the distribution of the tardiness and flow time to decrease compared to the other methods, enabling an improved estimation of lead times in the machine shop, which often is difficult in high-mix-low-volume settings.

Future research work includes dynamically learning workload norm levels based on shop-floor conditions, incorporating dynamic events within the system, and increasing the level of feedback from other agents to further improve the pre-shop sequencing rules.

REFERENCES

Bertolini, M., M. Braglia, M. Frosolini, and L. Marrazzini. 2020. "Work In Next Queue CONWIP". *Computers and Industrial Engineering* 143(August 2019):106437.

Bertrand, J. W. M., and A. M. Van De Wakker. 2002. "An Investigation of Order Release and Flow Time Allowance Policies for Assembly Job Shops". *Production Planning and Control* 13(7):639–648.

Bertrand, J. W. M., and H. P. Van Ooijen. 2002. "Workload Based Order Release and Productivity: A Missing Link". *Production Planning and Control* 13(7):665–678.

Branke, J., T. Hildebrandt, and B. Scholz-Reiter. 2015. "Hyper-Heuristic Evolution of Dispatching Rules: A Comparison of Rule Representations". *Evolutionary Computation* 23(2):249–277.

Darlington, J., M. Francis, P. Found, and A. Thomas. 2015. "Design and Implementation of a Drum-Buffer-Rope Pull-System". *Production Planning and Control* 26(6):489–504.

Didden, J. B., Q. V. Dang, and I. J. Adan. 2021, oct. "A Semi-decentralized Control Architecture for High-MIX-Low-Volume Factories in Industry 4.0". *Manufacturing Letters* 30:11–14.

Ebufegha, A., and S. Li. 2021. "Multi-Agent System Model for Dynamic Scheduling in Flexible Job Shops". In *Proceedings of the 2021 Winter Simulation Conference*, 10–27.

Gómez Paredes, F. J., M. Godinho Filho, M. Thürer, N. O. Fernandes, and C. J. C. Jabbour. 2020. "Factors for Choosing Production Control Systems in MakE-to-Order Shops: A Systematic Literature Review". *Journal of Intelligent Manufacturing* 33(3):639–674.

Holthaus, O., and C. Rajendran. 1997. "Efficient Dispatching Rules for Scheduling in a Job Shop". *International Journal of Production Economics* 48(1):87–105.

Kim, N., S. Barde, K. Bae, and H. Shin. 2021. "Learning Per-Machine Linear Dispatching Rule for Heterogeneous Multi-Machines Control". *International Journal of Production Research* 0(0):1–21.

Kingma, D. P., and J. L. Ba. 2015. "Adam: A Method for Stochastic Optimization". *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*:1–15.

Maoudj, A., B. Bouzouia, A. Hentout, A. Kouider, and R. Toumi. 2019. "Distributed Multi-Agent Scheduling and Control System for Robotic Flexible Assembly Cells". *Journal of Intelligent Manufacturing* 30(4):1629–1644.

Mourtzis, D. 2016. "Challenges and Future Perspectives for the Life Cycle of Manufacturing Networks in the Mass Customisation Era". *Logistics Research* 9(1):1–20.

Neuner, P., and S. Haeussler. 2021. "Rule Based Workload Control in Semiconductor Manufacturing Revisited". *International Journal of Production Research* 59(19):5972–5991.

Poslad, S. 2007. "Specifying Protocols for Multi-Agent Systems Interaction". *ACM Transactions on Autonomous and Adaptive Systems* 2(4).

Prakash, J., and J. F. Chin. 2014. "Modified CONWIP Systems: A Review and Classification". *Production Planning and Control* 26(4):296–307.

Ruder, S. 2016. "An Overview of Gradient Descent Optimization Algorithms". 1–14.

Sahin, C., M. Demirtas, R. Erol, A. Baykasoğlu, and V. Kaplanoğlu. 2017. "A Multi-Agent Based Approach to Dynamic Scheduling with Flexible Processing Capabilities". *Journal of Intelligent Manufacturing* 28(8):1827–1845.

Thürer, M., N. O. Fernandes, M. Stevenson, and T. Qu. 2017. "On the Backlog-Sequencing Decision for Extending the Applicability of Conwip to High-Variety Contexts: An Assessment by Simulation". *International Journal of Production Research* 55(16):4695–4711.

Thürer, M., M. J. Land, M. Stevenson, L. D. Fredendall, and M. Godinho Filho. 2015, jul. "Concerning Workload Control and Order Release: The Pre-Shop Pool Sequencing Decision". *Production and Operations Management* 24(7):1179–1192.

Thürer, M., C. Silva, M. Stevenson, and M. Land. 2012. "Improving the Applicability of Workload Control (Wlc): the Influence of SequencE-dependent Set-Up Times on Workload Controlled Job Shops". *International Journal of Production Research* 50(22):6419–6430.

Thürer, M., and M. Stevenson. 2018. "On the Beat of the Drum: Improving the Flow Shop Performance of the Drum–Buffer–Rope Scheduling Mechanism ". *International Journal of Production Research* 56(9):3294–3305.

Thürer, M., M. Stevenson, and C. W. Protzman. 2016. "Card-Based Production Control: A Review of the Control Mechanisms Underpinning Kanban, ConWIP, POLCA and COBACABANA Systems". *Production Planning and Control* 27(14):1143–1157.

Thürer, M., M. Stevenson, and T. Qu. 2016. "Job Sequencing and Selection within Workload Control Order Release: An Assessment by Simulation". *International Journal of Production Research* 54(4):1061–1075.

Thürer, M., M. Stevenson, C. Silva, and G. Huang. 2012. "The Application of Workload Control in Assembly Job Shops: An Assessment by Simulation". *International Journal of Production Research* 50(18):5048–5062.

Wang, S., J. Wan, D. Zhang, D. Li, and C. Zhang. 2016. "Towards Smart Factory for Industry 4.0: A Self-Organized Multi-Agent System with Big Data Based Feedback and Coordination". *Computer Networks* 101:158–168.

Weng, M. X., Z. Wu, G. Qi, and L. Zheng. 2008. "Multi-Agent-Based Workload Control for MakE-to-Order Manufacturing". *International Journal of Production Research* 46(8):2197–2213.

Zhou, Y., J. J. Yang, and L. Y. Zheng. 2019. "Multi-Agent Based Hyper-Heuristics for Multi-Objective Flexible Job Shop Scheduling: A Case Study in an Aero-Engine Blade Manufacturing Plant". *IEEE Access* 7:21147–21176.

AUTHOR BIOGRAPHIES

Jeroen B.H.C Didden is an PhD candidate in the section Operation, Planning, Accounting and Control, Department of Industrial Engineering and Innovation Sciences at the Eindhoven University of Technology (TU/e). His current research focuses on the application of Multi-Agent System to solve complex scheduling problems in High-Mix-Low-Volume production environments. His e-mail address is j.b.h.c.didden@tue.nl.

Q.V. Dang is an assistant professor in the section Operation, Planning, Accounting, and Control (OPAC), Department of Industrial Engineering and Innovation Sciences at the Eindhoven University of Technology (TU/e). Dang holds, since 2014, his PhD in Operations Research from Aalborg University in Denmark. Dang is passionate on developing methodologies for planning and scheduling automated (manufacturing/services) systems, including the interconnection between manufacturing processes and material handling operations and the collaboration between autonomous mobile robots and human operators. Email: q.v.dang@tue.nl

Ivo J.B.F. Adan is a Professor at the Department of Industrial Engineering of the Eindhoven University of Technology. His current research interests are in the modeling and design of manufacturing systems, warehousing systems and transportation systems, and more specifically, in the analysis of multi-dimensional Markov processes and queueing models. His email address is i.adan@tue.nl.