EVALUATION CONCEPT FOR ORDER RELEASE BASED ON SIMULATION AND REINFORCEMENT LEARNING AGAINST CHANGES TO THE PRODUCTION CONFIGURATION

Tim Janke[a], Michael Riesener[a], Seth Schmitz[a], Judith Fulterer[a], Hendrik Eisbein[a]

[a]Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Germany
{t.janke,m.riesener,s.schmitz,j.fulterer,h.eisbein}@wzl.rwth-aachen.de

ABSTRACT

Manufacturing an expanding number of diverse product variants while also enabling rapid responses to changes in the production schedule requires flexible production structures such as job shop production. Managing the resulting multitude of heterogeneous material flows leads to high complexity in production control. Hence, advanced dynamic scheduling methods based on simulation and reinforcement learning (RL) agents are required. These systems are confronted with changes in the production environment and unplanned disruptions, forcing the agent to handle deviated applications that require an adjustment. This paper presents an evaluation concept for a necessary retraining of production scheduling using a powerful control interface. The concept is based on the determination of an evaluation logic using logistic target values. By systematically analyzing changes in the production configuration in detail, respective production scenarios are compared with each other, deriving decision rules for the requisite retraining of the agent.

Keywords: job shop production, reinforcement learning (RL), production planning and control, advanced scheduling methods.

1 INTRODUCTION

Production control is considered one of the most relevant operational problems in production and its importance becomes recognizable once again when advanced remanufacturing processes complement conventional production to increase profits or in terms of environmental regulations [1, 2]. In contrast to production planning, which designs the general production content and processes, production control manages the actual order processing and sequencing [3]. One specific production control task is order release, which transfers orders from the planning level into the production, controlling the amount and the selection of orders [4]. From this point on, the planning is applied in the operative system.

Currently flexible production structures, e.g. job shop production, are essential to produce an increasing number of different product variants while at the same time being able to quickly react to changes in the production program [5, 6]. However, individual production processes for different components lead to a high number of inhomogeneous material flows [7]. In particular, the running production must be efficiently controlled, while at the same time, production planning must ensure that the right orders in the right sequence support the subsequent work plans [8]. Consequently, these requirements must be supported by advanced scheduling methods, where the task of order release shows a high influence on the planning quality to obtain high adherence to delivery dates. To cope with the challenges, dynamic methods combining simulation to suitably model complex production relations and reinforcement learning (RL) to learn specific strategies are a growing research field for solving production scheduling. [9]

RL approaches require a training phase, in which the RL algorithm interacts with a specific simulation environment [10]. However, production is confronted with changes in the production configuration and unplanned disruptions like machine breakdowns or personnel shortages. As a result, the underlying conditions in the running production system may increasingly deviate from those in a training phase. In practice, these disruptions are met by manual reprioritization and editing of running orders [11], while for

Proc. of the 2024 Annual Simulation Conference (ANNSIM'24), May 20-23, 2024, American University, DC, USA
C. Ruiz-Martin, B. Oakes and R. Cárdenas, eds.

approaches based on RL, it remains unclear if an already trained agent still meets the requirements or whether extensive retraining is required. Therefore, to ensure a high quality of the order release decision under occurring deviations, the algorithm's result must be continuously monitored to it must be decided when and to what extent an RL algorithm needs to be retrained in an appropriately adapted simulation environment.

Therefore, this paper presents an evaluation concept to decide, which occurring changes in a production system result in a necessary retraining of an RL agent used for production control. This concept is based on the determination of an evaluation logic using logistic target values. A systematic analysis of changes in the production configuration serves as a basis for comparing production scenarios with each other to derive decision rules for the requisite retraining. In addition, this approach integrates a powerful control and evaluation interface. This work is organized as follows: In section 2, the theoretical background of RL and its application in production planning and control is given. A review of order release strategies considering classic and AI-based approaches is presented in section 3. The evaluation concept for agent monitoring is described in section 4, including the control and evaluation interface and a validation of the metric. Finally, in section 5 the work is concluded and important aspects for further research are declared.

2 THEORETICAL BACKGROUND

In this section, the general principle of RL algorithms is introduced by explaining the problem formulation and the main elements. Then, their application as a tool for production planning and control is motivated and brought into a larger context.

2.1 Introduction and Functionality of Reinforcement Learning

RL is one main machine learning method that, in contrast to supervised or unsupervised learning, integrates feedback from its environment into the learning cycle [12]. The aim is to identify appropriate solutions for sequential decision problems by maximizing a cumulative reward function. In particular, RL algorithms solve discrete-time Markov Decision Problems (MDP), in which a future state only depends on the combination of a state and an action performed a time step before. MDPs consist of four parts – a state space representing the current status of an environment, an action function defining possible actions for each state, a transition function that describes the shift from one to another state and a reward function modeling an expected reward. [13] The interaction procedure between an RL algorithm, called the *agent*, and its counterpart the *environment*, often depicted by a simulation model, is shown in Figure 1.

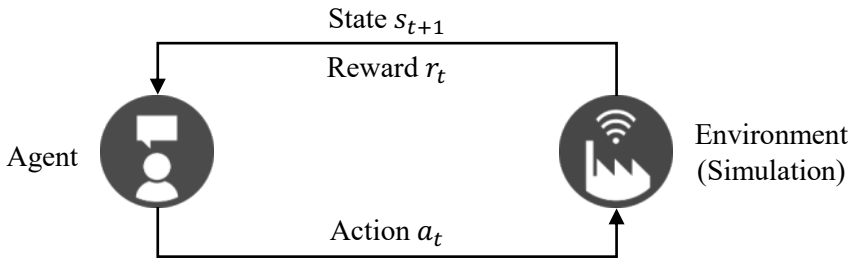

Figure 1: Principle of a reinforcement learning algorithm [14, 15].

At each discrete time step t, the agent observes the environment, represented by its state $s_t \in S$, where S is a set of possible states (state space). Based on that observation, the agent decides on an action $a_t \in A$, where A is a set of possible actions (action space), which causes the environment to switch its state from s_t into s_{t+1}. This transaction results in a specific reward r_t received by the agent, which will be maximized over time into a total reward $R_t = \sum_{i=t}^{T} r_i$ based on the specific objective. [15] During this procedure, the agent refines its policy $\pi_t(a|s)$ based on the actions it performs on specific states and what reward it gets from those actions [13].

To apply this principle to a specific problem, the state and action space as well as the reward function need to be formulated on an appropriate level of detail. Time steps can be fixed or depending on specific events, actions can range from low-level control of one parameter to high-level decisions and states as well as variate from one sensor signal to the states of a complex system. Then, the reward function, which depends on the design of the other two sections, ranges from simple objectives maximizing the total makespan to complex pre-processing procedures including domain knowledge and additional constraints.

Although a simulation model is typically used to represent the environment, RL algorithms can be distinguished by their interaction with this model. The algorithm is model-based when the simulation model itself is used to better understand the problem and forecast the next possible states. Model-free algorithms cannot access the model itself. They observe the values representing a state, choose an action based on it and receive a reward. This corresponds to a trial-and-error principle to learn an appropriate strategy. [15]

2.2 Application of Reinforcement Learning in Production Control

Applying RL-based approaches in production planning and control (PPC) is a state-of-the-art research field [9, 16] and has been considered in previous papers [14, 17]. This section introduces RL used in PPC in a broader context while section 3 looks into specific RL approaches in more detail. As explained in section 1, this is particularly true for flexible production structures such as job shop production.

The job shop scheduling problem (JSP) is a mathematical problem formulated precisely to solve production control tasks. For a given set of orders and machines, it is determined when which order is to be processed on which machine with regard to the optimal fulfillment of specified company goals. [14] By its definition, the JSP allows it to be formulated as MDP [18] and hence to be solved with the method of RL. Model-free RL approaches in particular have proven to be suitable for JSP as not always parameterized expert knowledge is available while a simulation model to simply represent the state space can be considered as a precondition [19]. Also because of its possibilities to be used in changing environments, a lot of attention has been paid to the application of RL to the JSP due to unpredictable adjustments and unplanned machine downtimes as a characteristic of production systems [16]. Those characteristics as well as a complex system architecture are symptomatically for manufacturing systems which is why the resulting process uncertainty has to be considered as a black-box problem [20].

In previous works, the authors identified three potential problems associated with a practical application of RL-based approaches in order release. First, a limitation in addressing problem sizes, with a predominant focus on small instances involving 3 to 15 machines, often avoiding authentic production data. Although the utilization of artificial or open-source data sets aids in meeting training data requirements, it falls short of providing solutions to real-world challenges. Additionally, the prevalent objective of minimizing makespan overlooks considerations such as adherence to delivery dates or capacity utilization. Finally, the absence of a well-defined application setup impedes the development of algorithms, lacking considerations for realistic shop floor conditions and a pronounced emphasis on logistical target values. [14]

3 RELATED WORK

This section reviews classic approaches and current work on the order release task. The related work is classified based on the used methods and for RL-based approaches, an overview of existing evaluation methods is given.

3.1 Literature Review on Order Release Methods and Categorization

For many years, order release methods have taken a major role in PPC literature due to continuous improvements and newly developed methods using advancements of mathematical techniques as well as computing resources. A set of classical approaches combined with heuristics and AI-based methods are differentiated.

3.1.1 Classic Approaches and Heuristics

A distinction is made between a total of four basic order release mechanisms, three of which can be regarded as real procedures and one contains the *immediate order release* directly after the order has been created [21, 22]. The other procedures are *scheduled order release*, *load-limited order release* and *inventory-regulating order release*. The *scheduled order release* (SOR) determines specific release dates per order at which they are released into production regardless of the utilization or inventory, e.g. backward scheduling [21, 23]. The release time can be determined in various ways, e.g. through given intervals, based on order or production information [24]. The second mechanism is *load-limited order release* or workload control (WC), which releases orders based on the current utilization of the shop floor without considering specific due date information [23]. Orders in this category can be divided into two dimensions: First, the aggregation of workstations determining the utilization value e.g. overall production, specific workstations or a bottleneck machine and second, the border used e.g. upper or lower limit [22]. The last mechanism comprises *inventory-regulating order release* methods, releasing orders when the number of orders in production that are either in the queue or being processed falls below or exceeds a predefined limit [21].

The proposed classification already includes methods, commonly referred to as heuristics, which try to find general rules rather than considering a specific state of production. Examples used in praxis are WC regarding the bottleneck machine and constant work in process (Conwip), an inventory-regulating method often indicating the referred inventory level behind, e.g. “Conwip 50” for 50 orders in process [21].

3.1.2 AI-based Approaches

With a focus on the respective objective of a learning-based system, the presented classical methods and heuristics have been extended by more advanced ones [25]. In particular, those approaches can be divided according to a data science perspective into artificial neural networks, fuzzy logic and evolutionary algorithms [9]. As comprehensive overviews of the most relevant learning-based approaches have been given already in previous papers [17], the most important ones are given in the following examples.

Most RL algorithms solving production scheduling tasks are based on model-free approaches and use single agents interacting with simplified PPC environments [26]. Others like [27], however, use a combined approach of dynamic scheduling for manufacturing components by combining the Monte Carlo Tree Search method, a multi-agent Deep Q-Network and a clustering of production orders in a comprehensive system architecture. Production orders are represented individually or as a group of similar orders by a separate DQN agent in order to enable a larger observation space. Then, the total lead time is optimized. [27]

Another recent approach covers a dispatching decision in a job shop production with multiple operations for each order, but the RL agent also uses Q-learning to fulfill its main objective of maximizing total lead time by deciding whether to use its self-determined sequence or a simple heuristic available for this selection. A validation against other dispatching rules has shown, that RL applications for dynamic scheduling become more beneficial when complexity increases in production, although a very small problem size has still been selected. [2] A larger problem size of eight machines using a comparable experimental setup has been introduced by [28]. Again, a single RL agent solves the dispatching problem in a job shop production minimizing average waiting times to shorten the lead time while keeping utilization high. As also found in the solutions introduced by [29] and [27], small problem size and the objective of minimizing total lead time is a regular application form leading to the shortcomings of practicability in terms of the industry need for higher adherence to delivery dates, because this reflects the actual perceived benefit [14, 26].

The majority of approaches use Q-learning in their method as identified in both the referenced literature reviews and the selected examples, but it is more frequently applied to the dispatching problem. However, since the order release decision already takes place in the preliminary stage of production control and, as described, has a major influence on adherence to delivery dates even before short-term disruptions have to be resolved by dispatching, the focus here remains on order release. The problem remains that small

quantities of production instances are used, which leads to the problem that the approaches solve the problems mathematically, but the scalability cannot yet be proven.

3.2 Evaluation Concept

Although several RL-based approaches include a validation of their method, only a few provide a comprehensive basis for decision-making regarding the context of changes in production between the training and application phases within an evaluation concept. These approaches mainly focus on the evaluation of *general requirements* covering the production environment and practicability as well as *order-specific requirements* mostly aiming at logistical target values. In addition, there exist approaches that make overarching comparisons and evaluations of order release strategies [30]. Finally, if new methods are applied to existing problems – such as RL in the context of PPC – those must be evaluated not only in terms of the result of the solved application but also in terms of the method itself. Therefore, *RL-specific requirements* must be included to evaluate the execution of the RL method: A basic measure is the ability of agents to learn a rewarding strategy in the context of simple decision problems. The efficiency of exploration measures how efficiently the agent can improve its strategy by trying out new actions. [31] The optimality gap is presented by [32] measuring the distance between the reward achieved by the agent and a specified minimum score. Finally, the duration of the decision-making process must be considered as an order release system must be able to quickly react in case of changes or disruptions.

The analysis of existing approaches has shown that they are suitable either for the evaluation of the decision result for the order release task or regarding the application of the RL method. An approach that combines both categories and establishes the connection between RL applications and order release procedures could not be found.

4 APPLICATION

This section introduces the RL agent used in this work and presents the methodology for monitoring the agent's decision quality and decision on necessary retraining efforts. Then, a development and evaluation interface is presented that allows for rapid validation.

4.1 Introduction of the Used Reinforcement Learning Agent and Simulation

On the basis of the RL agent introduced by [33], it has been further improved by [14] (see Figure 2), solving shortcomings addressed in section 2.2 – problem size, objective regarding adherence to the delivery date and practical implementation.

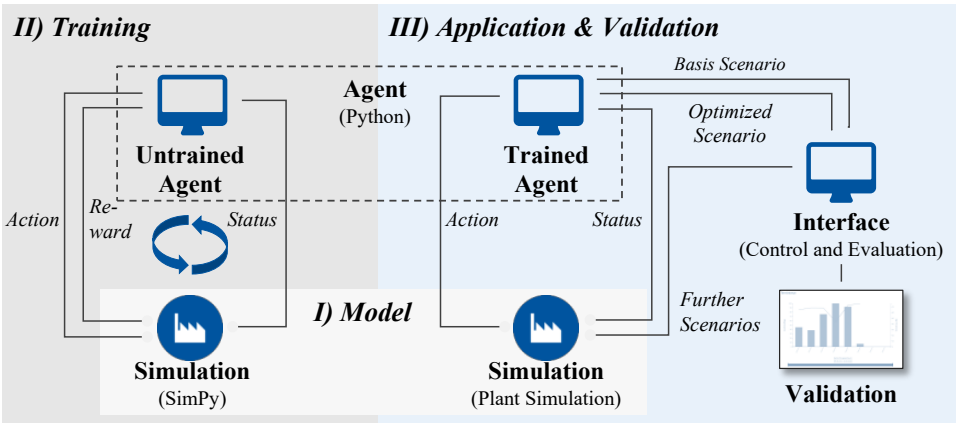

Figure 2: Used simulation and RL set-up [14].

The approach uses two discrete-event material flow simulation models on a machine and order basis. One simplified simulation in SimPy is used for the training phase (see stage II) modeling the main production

resources and relationships for the training phase to keep the data transfer between simulation and agent in training within the python environment. The application phase then switches to a more realistic simulation in Plant Simulation (see stage III) including statistically distributed machine breakdowns and order cancellations, the agent must deal with. This also requires a socket connection between the agent and the simulation model, which links the two components and enables communication. Using this kind of socket connection leads to a higher time requirement for training and testing. As a result, an average decision step in the Plant Simulation set-up takes approximately 25 times longer than in the SimPy environment. As the focus has been set on order release, for the operating sequence of each machine the first-in-first-out (FIFO) rule is implemented in both models as a simplified assumption. In this example, the job shop production of a component manufacturer is modeled in both simulations (see section 4.3).

The state space s_t is represented by the state vector providing the agent with information to decide on its next action. Here, work plans and machine lists are initially loaded into the simulation environment, whereas machine and order status renew every time step: General information comprises the *current episode* and *simulation time*. The machine status includes *availability*, *remaining processing time of queue* and *current order*. The order status stores the *allocated* and *downstream machines* as well as *processing times* and *due dates*. Dependent action space is selected here, where jobs are directly chosen to be released or not released at each time step [14] instead of approximating durations as [33]. An order pool is introduced, initially filled with pending orders and prioritized by the due date, where each action has an index corresponding to the index of an order. In addition, a "no-op" action is introduced, representing the possibility of not releasing an order. The reward function focuses on maximizing *adherence to delivery dates* by using the difference between the *remaining time until the due date* and the *remaining process time* [14].

4.2 Methodology for Agent Monitoring

One disadvantage of learning-based approaches is that the strategy learned cannot be formalized, or only with great effort, which means that the decision-making process remains non-transparent [34]. In the event of deviations in the agent's performance, assessing causes is difficult. One possible cause is a discrepancy between the initial training environment and the current application. As real productions are subject to dynamic changes these deviations must be continuously monitored to assess whether the agent needs to be re-trained. The methodology comprises an evaluation system that quantifies the quality of the decision outcome of an RL agent and derives rules to assess whether retraining is necessary. It is based on existing models (see section 3.2) and resolves the identified shortcoming by evaluating key figures regarding production and logistics as well as methodically relevant key figures on the application of RL algorithms. The four steps of the methodology including a continuous control loop is shown in Figure 3.

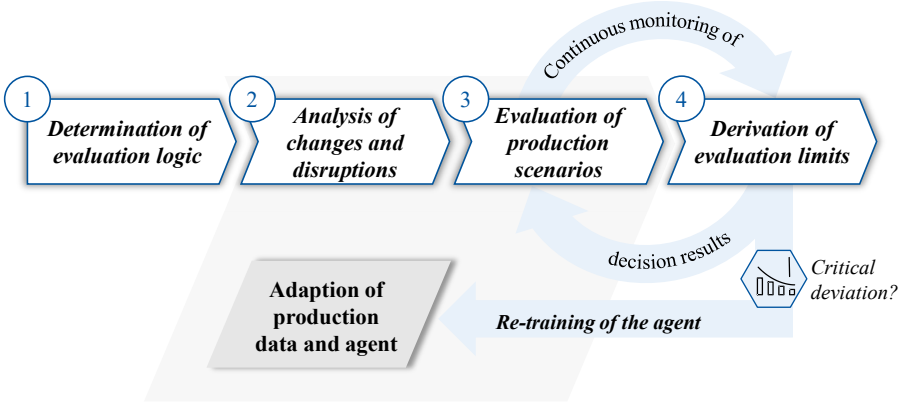

Figure 3: Methodology to evaluate the quality of order release decisions.

4.2.1 Determination of the Evaluation Logic based on Logistic Target Values

The first step involves determining key figures to evaluate the performance of the approach against conventional order release methods and specifications of the RL algorithm. The logistical target values according to [35] and [36] are used as key production figures: *Lead time*, *adherence to delivery time*, *order delay* and *capacity utilization*. The *lead time* is a central objective of a successful PPC. It measures the average time between the release and completion of an order and has a direct influence on on-time delivery as well as the accumulation of inventories [21]. Important, market-related criteria include the *adherence to delivery date* indicating the percentage of delayed orders (in case of a difference of more than one day) and the *order delay* calculated from the root mean square of delay in days of all delays orders. By using the quadratic mean, the distribution of the delay is taken into account to provide a more reliable value for the average delay. *Utilization* is an important key figure for evaluating the capacity and production efficiency It indicates the percentage use of production resources and sets the actual output about the target output. With an increasing number of variants achieving high machine *utilization* becomes more difficult, so it must be included in the consideration.

The two indicators *optimality gap* [32] and *decision duration* are selected to evaluate the algorithm figures. The *optimality gap* measures the distance between the reward achieved by the agent in a test versus the specified minimum score achieved when testing the agent in the original production environment. In addition, the *decision duration* plays an important role in production practice as it must be quickly able to react to changed conditions. It measures the time an agent needs to solve a complete simulation problem.

4.2.2 Analysis of Changes and Deviations in the Production Configuration

In the second step, the framework conditions that influence the production configuration are derived from the literature [37]. A distinction is made between *planned changes* in production and *unplanned deviations* that lead to a difference between the training and application environment. Both are divided into the categories of production process, order and production resource as listed in Table 1.

Table 1: Overview on planned changes and unplanned deviations.

Type	Category	Parameter
Planned changes	Production process	Changed sequence rule
	Order	Set-up and processing time change
	Production resource	Number of similar machines
		Machine production time
		Machine operating days
Unplanned deviations	Production process	Quality problem
		Incorrect logistic process
	Order	Product, demand time, volume change
		Prioritized, cancelled order
		Material availability
	Production resource	Machine breakdown
		Personnel availability

Planned changes describe intended adjustments, such as the sequence rule, that can be deliberately changed for each machine to change the processing queue. Set-up and processing times can be adjusted in the course of process optimizations or new tools and impact the total completion time. The last group of planned changes relates to production resources, where the number of similar machines, the machine production time and the machine operating days can change. *Unplanned deviations* are disruptions that affect the order

release process, are unintentional and can occur suddenly. Here, quality problems are included as dominant causes of disruptions and reworking activities. Incorrect logistic processes, e.g. incorrect scheduling of materials and tools, incorrect order picking or a lack of internal logistics vehicles, also negatively influence the production flow. Moreover, orders are subject to changes, e.g. added urgent orders that require prioritized processing or canceled orders that must be deleted from the planning basis. Product changes may lead to completely different production processes. Finally, machine breakdowns and understaffing can occur for various reasons and last for uncertain length reducing the production capacity.

4.2.3 Creation and Evaluation of Respective Production Scenarios

The third step involves the actual evaluation of the decision results, in which scenarios are simulated, evaluated and compared to each other. Based on the collection of changes and deviations in 4.2.2, problems that occur in a given production can be aggregated into a new scenario (adapted scenario). In addition to the initial scenario and the adapted one, a third scenario is built, which contains the same changes but uses a different order release method (comparative scenario). For this purpose and in the considered scenario, Conwip 50, scheduled order release (SOR) based on backward scheduling and workload control (WC) (see section 3.1.1) have shown the best results and are consequently chosen, whereby for each parameter the best-performing method is selected.

If there is a deviation in the agent's performance in the second scenario, detailed individual evaluations are necessary to identify the main impact. To carry this out, further scenarios are generated that consider each change or disruption individually. For the individual scenarios, the orders are released once by the agent and once by the best comparative method. There are then two scenarios (agent and comparative method) for each change and deviation. To minimize the effort involved, the individual scenarios are not generated upfront, but only on demand using the decision hierarchy in Figure 4. Critical deviations are assessed according to each key figure, which are detailed in the next section.

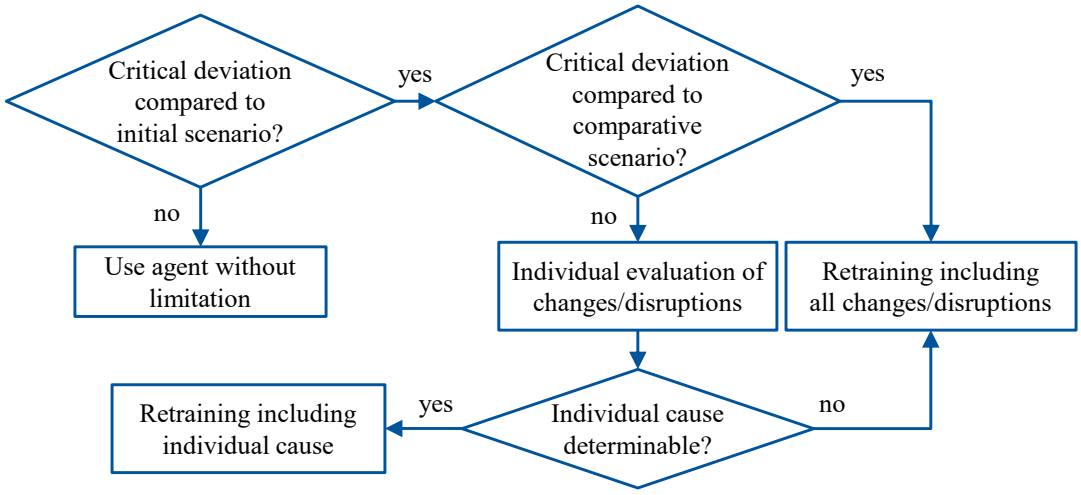

Figure 4: Overview of the decision hierarchy.

4.2.4 Derivation of Evaluation Limits and Decision Rules for Retraining of the Agent

Finally, in the fourth step, decision rules are given to evaluate critical deviation in the decision results and to what extent retraining is required. When monitoring the decision results of an agent it is important to decide on evaluation limits. To ensure an objective assessment evaluation thresholds of five percent are set within a three-stage system [38]. The use of a three-stage system has the advantage of forming a middle interval to register an existing deviation without being classified as critical. The deviation for each key figure of the comparative scenario is determined and assigned to one of three categories: No negative

deviation (1), a relative negative deviation smaller than five percent (2) and a critical relative negative deviation higher than five percent (3).

The classification into three categories is carried out individually for each of the six evaluation criteria and then aggregated into decision rules to make a holistic evaluation. Since a high-quality decision result can only be achieved by holistic fulfillment of all criteria, these are weighted equally. The decision hierarchy (see Figure 4) is used to determine the extent of necessary retraining. Based on the comparisons between the agent's result and those of a comparative method, the extent to which retraining is necessary is identified. The degree of retraining is divided into complete retraining and retraining of individual changes, keeping retraining effort to a minimum.

4.2.5 Integrated Control and Evaluation Interface

In addition to the four main action steps, the developed methodology has a control loop, which is created by repeating steps 3 and 4. This is intended to ensure a consistently high quality of the decision result. The control loop is embedded into the control and evaluation interface presented by [7], which supports automated pre-processing of the raw production data, manages different projects and scenarios and provides an integration into the simulation and python environments to handle all instances in just one tool. Thus, important simulation support such as pre-assignment of scenarios to fill production with existing orders in a realistic way and simulation of experiment instances for visualization are integrated into the tool. Finally, comparing the different scenario types by their respective key figures helps to quickly decide on necessary decisions for a running order release system.

4.3 Validation of the Metric

For validation, the knowledge gained in steps 1 and 2 of the methodology is used to create an adapted test scenario from the initial scenario. A job shop production scenario including ten machine types and a total of 76 orders to be released within the simulation time of three months is used as an initial scenario. Again, a FIFO principle is applied as the default sequencing rule before each machine. The discrete-event simulation then models the flow of the orders that already preoccupy the machines and orders that are released within simulation time. The changes made in the adapted scenario are aggregated in Table 2. To include unplanned disruptions, the simulation environment Siemens Plant Simulation provides the possibility of statistically distributed disruptions such as machine breakdowns modeled through reduced resource availabilities.

Table 2: Considered adaptions to the initial scenario.

Type	Parameter	Initial scenario	Adapted scenario
Planned	Sequence rule	FIFO	LIFO
	Set-up and processing time		Selected modifications
	Number of similar machines		3 machine types reduced by 1-2 instances
	Machine production time		Changed shift system
	Machine operating days		Changed availability times
Unplanned	Machine breakdown	100% availability	Reduced availability

Then, by applying the agent trained in the initial scenario to the adapted scenario, the following deviations regarding the key figures can be observed (see Table 3). The adapted scenario shows a critical deviation in three key figures (*lead time, utilization and order delay*). In addition, a deviation below the threshold value of 5% is recognizable for the *adherence to the delivery date* and *optimality gap*.

Table 3: Comparison of basis and adapted scenario.

Key figure (opt. direction)	**Initial scenario**	**Adapted scenario**	**Deviation**	**Comparative scenario**	**Deviation**
Lead time [d], ↓	4.1	5.7	+ 39.0%	4.3 (WC)	+ 4.9%
Utilization [%],↑	29.1	24.6	- 15.5%	34.8 (WC)	+ 19.6%
Adh. to deliv. date [%], ↑	86.8	85.5	- 1.5%	88.2 (WC)	+ 1.6%
Order delay [d], ↓	9.6	10.3	+ 7.3%	10.1 (WC)	+ 5.2%
Optimality gap [%], ↑	0%	-0.9%	- 0.9%	/	/
Decision time [s], ↓	31.0	31.0	+/- 0.0%	/	/

According to the decision hierarchy (see Figure 4), a critical deviation in at least one evaluation criterion requires the comparison against comparative scenarios based on different order release methods. In this configuration, WC has shown the best results for the respective figures. Industry standards for this kind of production for the utilization range around 50%, for adherence to delivery date between 80% and 90% and order delay between three to ten days. Lead time can be best benchmarked with the net operation times which range between a few minutes and a few hours and cumulate differently according to the lot size and shift model. According to the recommendations, complete retraining is necessary in this case, taking into account all changes and disruptions, as three of six key figures are considered critical for both the adapted scenario against the initial scenario and against comparative scenarios.

5 CONCLUSION AND FURTHER RESEARCH

This paper elaborates on an evaluation concept for learning-based order release systems to decide if an underlying RL agent requires retraining in the course of changes to the production configuration. It presents a four-step methodology for monitoring the decision quality using key figures regarding production and algorithm performance. After introducing the problem definition of job shop scheduling and identifying shortcomings of current evaluation approaches for RL-based production control approaches, the methodology is described in detail and validated in a real production example using a comprehensive development and evaluation tool.

The methodology solves the problem that real productions are usually confronted with continuous changes and unplanned disruptions to which specific, learning-based approaches cannot react without adjustment. In particular, it evaluates whether and to what extent retraining in an adapted initial scenario is necessary. By systematically analyzing possible changes and disruptions in the production configuration, comparative scenarios are generated and evaluated against each other, deriving specific decision rules for retraining of a RL agent. These range between a complete retraining process, minor adjustments in the training scenario or the decision that no retraining is necessary. The focus of further research remains on a practical formulation of the reward function and action space of considered RL agents in order release. This makes it possible to be used for realistic production sizes and to efficiently outperform existing methods. Also to meet the challenge that learned strategies can only be formalized with great effort, it is an accomplishment to derive rules from the behavior of a RL agent to enrich production knowledge.

ACKNOWLEDGMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany‘s Excellence Strategy – EXC-2023 Internet of Production – 390621612

REFERENCES

[1] H. Gunasekara, J. Gamage, and H. Punchihewa, "Remanufacture for Sustainability: A review of the barriers and the solutions to promote remanufacturing," in *2018 International Conference on Production and Operations Management Society (POMS)*, Peradeniya, Sri Lanka, pp. 1–7, 2018.

[2] C. Kardos, C. Laflamme, V. Gallina, and W. Sihn, "Dynamic scheduling in a job-shop production system with reinforcement learning," *Procedia CIRP*, vol. 97, pp. 104–109, 2021.

[3] G. Schuh, *Produktionsplanung Und -Steuerung 1: Grundlagen der PPS,* 4th ed. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2012.

[4] S. Haeussler, C. Stampfer, and H. Missbauer, "Comparison of two optimization based order release models with fixed and variable lead times," *International Journal of Production Economics*, vol. 227, pp. 1–17, 2020.

[5] N. Duffie, J. Bendul, and M. Knollmann, "An analytical approach to improving due-date and lead-time dynamics in production systems," *Journal of Manufacturing Systems*, vol. 45, 2017.

[6] H. Zijm, M. Klumpp, A. Regattieri, and S. Heragu, Eds., *Operations, Logistics and Supply Chain Management*. Cham: Springer International Publishing, 2019.

[7] G. Schuh, S. Schmitz, J. Maetschke, T. Janke, and H. Eisbein, "Intuitive Analyse komplexer Materialflüsse," *wt*, vol. 113, no. 04, pp. 171–175, 2023.

[8] G. Schuh, J.-P. Prote, A. Gützlaff, K. Thomas, F. Sauermann, and N. Rodemann, "Internet of Production: Rethinking production management," in *Production at the leading edge of technology*, J. P. Wulfsberg, W. Hintze, and B.-A. Behrens, Eds., Berlin: Springer, pp. 533–542, 2019.

[9] S. Lang, M. Schenk, and T. Reggelin, "Towards Learning- and Knowledge-Based Methods of Artificial Intelligence for Short-Term Operative Planning Tasks in Production and Logistics: Research Idea and Framework," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2716–2721, 2019.

[10] P. Theumer, F. Edenhofner, R. Zimmermann, and A. Zipfel, *Explainable Deep Reinforcement Learning for Production Control*: Hannover : publish-Ing, 2022.

[11] O. Lohse, S. Krause, C. Saal, and C. Lipp, "Real Time Reaction Concept for Cyber Physical Production Systems," in *2020 3rd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS)*, Gjovik, Norway, pp. 1–5, 2020.

[12] A. V. Joshi, *Machine Learning and Artificial Intelligence,* 1st ed. Cham: Springer, 2020.

[13] C. Watkins, "Learning from Delayed Rewards," Dissertation, Royal Holloway, University of London, London, 1989.

[14] G. Schuh, S. Schmitz, J. Maetschke, T. Janke, and H. Eisbein, *Application of a Reinforcement Learning-based Automated Order Release in Production*: Hannover : publish-Ing, 2023.

[15] R. S. Sutton and A. Barto, *Reinforcement learning, second edition: An introduction*. Cambridge, Massachusetts, London, England: The MIT Press, 2018.

[16] M. Panzer, B. Bender, and N. Gronau, *Deep Reinforcement Learning In Production Planning And Control: A Systematic Literature Review*: Hannover : publish-Ing, 2021.

[17] M. Kemmerling, V. Samsonov, D. Lütticke, G. Schuh, A. Gützlaff, M. Schmidhuber, and T. Janke, "Towards Production-Ready Reinforcement Learning Scheduling Agents: A Hybrid Two-Step Training Approach Based on Discrete-Event Simulations," in *Simulation in Produktion und Logistik 2021*, J. Franke and P. Schuderer, Eds., Göttingen: Cuvillier Verlag, pp. 325–336, 2021.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," (eng), *Nature*, vol. 518, no. 7540, pp. 529–534, 2015.

[19] J. Xie, L. Gao, K. Peng, X. Li, and H. Li, "Review on flexible job shop scheduling," *IET Collaborative Intelligent Manufacturing*, vol. 1, no. 3, pp. 67–77, 2019.

[20] M. Zhang, "Resource allocation problems in manufacturing systems using white-box-simulation-based cut generation approach," Dissertation, Politecnico di Milano, Milano, 2021.

[21] H. Lödding, *Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration,* 3rd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[22] N. O. Fernandes and S. do Carmo-Silva, "Generic POLCA—A production and materials flow control mechanism for quick response manufacturing," *International Journal of Production Economics*, vol. 104, no. 1, pp. 74–84, 2006.
[23] D. Bergamaschi, R. Cigolini, M. Perona, and A. Portioli, "Order review and release strategies ina job shop environment: A review and a classification," *International Journal of Production Research*, vol. 35, no. 2, pp. 399–420, 1997.
[24] I. Sabuncuoglu and H. Y. Karapınar, "Analysis of order review/release problems in production systems," *International Journal of Production Economics*, vol. 62, no. 3, pp. 259–279, 1999.
[25] A. Jones, L. C. Rabelo, and A. T. Sharawi, "Survey of Job Shop Scheduling Techniques," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed., Hoboken, NJ, USA: John Wiley & Sons, Inc, 2001.
[26] A. Esteso, D. Peidro, J. Mula, and M. Díaz-Madroñero, "Reinforcement learning applied to production planning and control," *International Journal of Production Research*, vol. 61, no. 16, 2023.
[27] A. Kuhnle, J.-P. Kaiser, F. Theiß, N. Stricker, and G. Lanza, "Designing an adaptive production control system using reinforcement learning," *J Intell Manuf*, vol. 32, no. 3, pp. 855–876, 2021.
[28] B. Waschneck, "Autonome Entscheidungsfindung in der Produktionssteuerung komplexer Werkstattfertigungen," Dissertation, 2020.
[29] O. Lohse, "Entwicklung einer Methode zum Einsatz von Reinforcement Learning für die dynamische Fertigungsdurchlaufsteuerung," Dissertation, KIT, Karlsruhe, 2023.
[30] V. R. Guide and R. Srivastava, "An evaluation of order release strategies in a remanufacturing environment," *Computers & Operations Research*, vol. 24, no. 1, pp. 37–47, 1997.
[31] I. Osband, Y. Doron, M. Hessel, J. Aslanides, E. Sezener, A. Saraiva, K. McKinney, T. Lattimore, C. Szepesvari, S. Singh, B. van Roy, R. Sutton, D. Silver, and H. van Hasselt, "Behaviour Suite for Reinforcement Learning," 2019.
[32] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[33] V. Samsonov, M. Kemmerling, M. Paegert, D. Lütticke, F. Sauermann, A. Gützlaff, G. Schuh, and T. Meisen, "Manufacturing Control in Job Shop Environments with Reinforcement Learning," in *International Conference on Agents and Artificial Intelligence*: SCITEPRESS - Science and Technology Publications, pp. 589–597, 2021.
[34] A. Sharma, K. Xu, N. Sardana, A. Gupta, K. Hausman, S. Levine, and C. Finn, "Autonomous Reinforcement Learning: Formalism and Benchmarking," Dec. 2021.
[35] H.-P. Wiendahl and H.-H. Wiendahl, *Betriebsorganisation für Ingenieure,* 9th ed. München: Hanser, 2019.
[36] P. R. Philipoom, M. K. Malhotra, and J. B. Jensen, "An Evaluation of Capacity Sensitive Order Review and Release Procedures in Job Shops," *Decision Sciences*, vol. 24, no. 6, 1993.
[37] V. R. Guide and R. Srivastava, "An evaluation of order release strategies in a remanufacturing environment," *Computers & Operations Research*, vol. 24, no. 1, pp. 37–47, 1997.
[38] C. Brell, J. Brell, and S. Kirsch, *Statistik von Null auf Hundert: Mit Kochrezepten schnell zum Statistik-Grundwissen,* 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.

AUTHOR BIOGRAPHIES

TIM JANKE (*1993) studied Mechanical Engineering (Production Engineering) at RWTH Aachen University in Germany and Industrial Engineering at Tsinghua University in Beijing. He is a Research Assistant in Production Logistics in the Production Management department at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University. His email address is t.janke@wzl.rwth-aachen.de.

MICHAEL RIESENER (*1986) is the Managing Chief Engineer of the Chair of Production Engineering at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University, is the managing director of the Center for Systems Engineering and Center Smart Industrial Agriculture as well

as member of the Advisory Board of the Center for Circular Economy at the RWTH Aachen University. His email address is m.riesener@wzl.rwth-aachen.de.

SETH SCHMITZ (*1991) studied Business Administration and Engineering (Mechanical Engineering) at the RWTH Aachen University and Tsinghua University. He is Head of the Production Management department at the Laboratory for Machine Tools and Production Engineering WZL and is the Managing Director of the Global Production Management Center at the RWTH Aachen University. His email address is s.schmitz@wzl.rwth-aachen.de.

JUDITH FULTERER (*1994) studied Business Administration and Engineering (Mechanical Engineering) at RWTH Aachen University in Germany. She is a Research Assistant at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University and Group Lead of the Production Logistics group in the Production Management department. Her email address is j.fulterer@wzl.rwth-aachen.de.

HENDRIK EISBEIN (*1998) studied Business Administration and Engineering (Mechanical Engineering) at the RWTH Aachen University in Germany. He is a Research Assistant in Production Logistics in the Production Management department at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University. His email address is h.eisbein@wzl.rwth-aachen.de.