4th Conference on Production Systems and Logistics

Application of a Reinforcement Learning-based Automated Order Release in Production

Günther Schuh[1,2], Seth Schmitz[1], Jan Maetschke[1], Tim Janke[1], Hendrik Eisbein[1]

[1]*Laboratory for Machine Tools and Production Engineering (WZL) of the RWTH Aachen University, Aachen, Germany*
[2]*Fraunhofer Institute for Production Technology IPT, Aachen, Germany*

Abstract

The importance of job shop production is increasing in order to meet the customer-driven greater demand for products with a larger number of variants in small quantities. However, it also leads to higher requirements for the production planning and control. In order to meet logistical target values and customer needs, one approach is the focus on dynamic planning systems, which can reduce ad-hoc control interventions in the running production. In particular, the release of orders at the beginning of the production process has a high influence on the planning quality. Previous approaches used advanced methods such as combinations of reinforcement learning (RL) and simulation to improve specific production environments, which are sometimes highly simplified and not practical enough. This paper presents a practice-based application of an automated order release procedure based on RL using the example of real-world production scenarios. Both, the training environment, and the data processing method are introduced. Primarily, three aspects to achieve a higher practical orientation are addressed: A more realistic problem size compared to previous approaches, a higher customer orientation by means of an objective regarding adherence to delivery date and a control application for development and performance evaluation of the considered algorithms against known order release strategies. Follow-up research will refine the objective function, continue to scale-up the problem size and evaluate the algorithm's scheduling results in case of changes in the system.

Keywords

Reinforcement learning; agent; automated order release; simulation; job shop production

1. Introduction

Due to a customer-driven increasing demand for higher product variants in correspondingly smaller numbers, flexible structures must enable their production and therefore the importance of job shop production is growing [1]. However, in a job shop production, the products take different routes through the production which complicates the allocation of machine capacities as well as operator and material availabilities to specific orders. To keep track of the order-specific view and simultaneously control the overall system, requirements on production planning and control (PPC) increase [2]. In order to cope with the higher requirements, one recognizable focus of PPC lies in the optimization of throughput times [3] while other logistical target values such as capacity utilization and adherence to delivery date still remain relevant for manufacturing companies [4]. In consequence, PPC processes become more dynamic and advanced [5].

Order-related optimization attempts can be achieved on two levels – on the upper level (order release) considering the logistical chain by starting production orders and on the lower level (sequencing) by

DOI: https://doi.org/10.15488/13500
ISSN: 2701-6277

changing the queue sequence in front of production units [3]. Although the high importance of an optimized order release system on the planning quality is well known [6], it is not yet used as standard today to achieve logistical target values and to reduce ad-hoc control interventions [7]. When focusing on scheduling tasks in a job shop production, discrete-event simulation is suitable for modelling the complex interrelationships and thus the main system behaviour [8]. Especially, in the context of flexible routes and considering unplanned machine downtimes, simulation holds a significant role in solving PPC tasks [9]. Previous paper and recent research on that topic combine simulation with reinforcement learning (RL) as an important field of machine learning [10]. By utilizing current advances in algorithm development with RL and combining this with simulation, a promising tool to effectively solve production scheduling problems is created [11].

In this context, recent approaches put a strong focus on further development from a computer science point of view and do not reflect the realistic complexity and framework conditions of real-world production systems in terms of machines, orders and uncertainties. It becomes important to align those approaches with practical requirements derived from an engineering perspective, e.g. a realistic problem size and representative data set [1]. Therefore, this paper formulates necessary steps for a stronger practical orientation of RL approaches in production scheduling based on the approach presented in [12]. The remaining of this work is organized as follows: In section 2, the studied problem is described, the method for implementing RL-based scheduling problems is specified and it is explained why previous approaches for order release are not yet practice-based enough. A review on order release strategies considering conventional approaches, heuristics and concepts based on RL is given in section 3. The application of a practice-based RL-approach for automated order release is explained in section 4. Finally, in section 5 the work is concluded and important aspects for further research are elucidated.

2. Background

In this section, the main reason for optimizing the representative task of order release is explained and the general principle of RL algorithms as a tool used for production scheduling is introduced. Then, with the rise of promising RL approaches, the need for more practice-based approaches is motivated.

2.1 Order release in the job shop scheduling problem (JSP)

In our previous paper [1], the substantial reasons for focusing on order release as a representative task within production control (PC) have been motivated. As argued, order release marks a "*critical decision point*" [13] for subsequent PC tasks and regardless a widespread use of enterprise resource planning (ERP), advanced planning and scheduling software (APS) [4] and still conventional heuristics, there is further need for optimization in production practice [1]. Especially due to the further increasing importance of job shop production, which has been proven to be NP-hard [5], it becomes necessary to develop advanced methods to solve the known practical problem.

Therefore, this work considers the order release in a job shop production by adapting the typical assumptions [1,14]:

- One operation at a time on each machine and on any job
- An operation of a job can be executed by only the assigned machine
- The next operation of a job can be started after completing its preceding operations
- No alternative routings for a job
- Each machine is available for production according to the machine calendar (in the application phase additional machine breakdowns and order cancellations will be included)
- No restriction on queue length before any machine

2.2 Implementing a RL algorithm for production scheduling

For solving job shop scheduling problems, either basic correlations presented as heuristics or deep problem related knowledge for exact solutions are required [15]. Since such knowledge is not always available and significant effort is required to parameterize expert knowledge, the desire for further advanced approaches has emerged [16]. Model-free RL has proven to be a suitable approach without requiring this expert knowledge [11] by just interacting with the production environment in a "trial and error" scheme [17].

In order to be used in a RL algorithm job shop scheduling problems are modelled as a Markov Decision Problem (MDP). MDPs are characterized by the fact that future states only depend on the present state and action [18]. Since the decision process requires frequent repetitions it is not directly applicable to real production environments and thus needs to be represented by a discrete-event simulation [8,19].

The core idea of reinforcement learning is based on the interaction between a RL algorithm – referred to as agent – and an environment – usually represented by a simulation model (see Figure 1). At each time step, the agent observes the environment described by its state $s_t \in S$ and decides on an action $a_t \in A$ to perform. Subsequently, the taken action leads to the state $s_{t+1} \in S$ and a specific reward for the agent, which over time aggregates to a total reward of $R_t = \sum_{i=t}^{T} r_i$. Those two aspects are combined during the training phase, where the agent tries to learn and adapt an optimal policy $\pi_t(a|s)$ based on the actions it performs and the reward it gets [20]. The learnt policy allows the agent in the following application phase to solve the stated problem. According to the defined objective function, during training phase the RL agent tries to maximize the sum of rewards gotten from its actions in order to solve the stated problem [21]. For the approach at hand, related works suggest the Deep Q-Network (DQN) training method as a suitable approach [11,12].

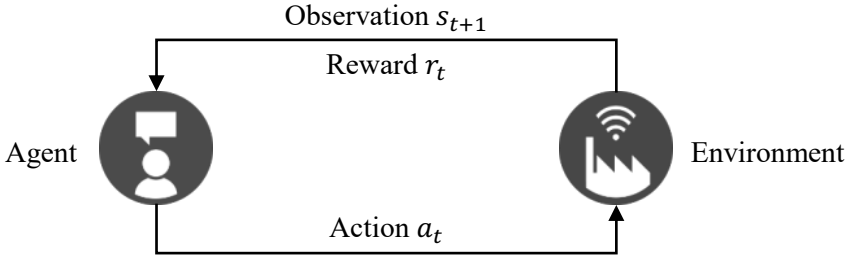

Figure 1: General principle of RL algorithms [20]

Especially in the training phase, the environment is represented by a simulation model, because in contrast to real systems, it reacts reproducibly and thus can be iterated [22]. For training, the agent needs the interface to the simulation model and a set of training data. Complying with the practice requirement claimed in this paper, this should be historical feedback data from an ERP system. For training purposes fast interfaces between the simulation model and the RL agent significantly improves the training time [11]. For the application after the completed training phase, the trained RL agent can apply its improved value function on a production case either still represented by a simulation or a real production.

2.3 Practical application of RL-based production scheduling

As stated in the introduction, this paper is intended to give a more practical direction to the application of RL algorithms in production scheduling. Therefore, the following three sections present what would be required for more practice-based approaches. Similar to other areas e.g. big data analysis, this field of research also benefits from cooperation between data science and engineering, because engineering brings in the necessary production expertise[23].

2.3.1 Problem size and data set

A major challenge for production scheduling approaches is the excessively increasing complexity with the increase in problem size until an approximate realistic production scale of e.g. 200 machines is reached. One problem of RL approaches that prevents user from simply scaling-up the problem size is the state design representing the number of jobs and machines. It can be overcome by using production expertise for state aggregation, function approximation or modelling adaptions in the state-action space [24]. Another requirement for practice-based approaches would be the validation with real production data to ensure transferability to real-life problems [1]. Without focus on features of real production environments, possible influencing factors such as machine breakdowns, order cancellations, sequence dependent setup time, and precedence constraints could thus be unintentionally hidden [24].

2.3.2 Objective function and action space

One important enabler of a performant RL approach is the objective, from which the reward is directly or indirectly derived. Yet, it is noticeable that the total makespan is the most widespread objective [24,11]. Although reliable results in not directly optimized performance measures are achieved as well, a survey of manufacturing companies showed, that due to recent crises, companies are paying more and more attention to on-time delivery as a target for their PPC [7]. Therefore, this paper claims to primarily focus on the value adding measure adherence to delivery date. Here it should be also covered that a new order release method should directly focuses on identification of orders instead of indirectly determining them.

2.3.3 Control interface for development and evaluation

The third missing aspect towards practice-based approaches is a specialized control interface for fast evaluation and algorithm development, by means of evaluation against logistical target values and comparison against classical or adapted control heuristics. Standard control and evaluation interfaces such as Tensorboard or WandB, that are widely used for ML development[25,26], mainly focus on the comparison of different algorithm variants and the performance evaluation of machine learning methods. Certainly, for optimizing the RL method itself, this is an important part of algorithm development, but it misses the special characteristics that must be mapped in production to justify a method even against other less advanced solution options that are not ML based. Therefore, the application presented in this paper integrates the Plant Simulation environment to easily create and compare different scenarios based on common order release heuristics or an optimized solution from the RL agent. With the selected combination of order and machine related graphs an effective analysis of a job shop production is enabled based on production expert knowledge.

3. Related work

In this section, conventional and recent approaches on the job shop scheduling problem are reviewed. Especially the task of order release and approaches with transferable findings are considered in detail. In the second part of the chapter, existing AI-based approaches for order release and order scheduling are specified.

3.1 Conventional approaches and heuristics

A brief overview on order release methods has been given in the previous paper [1]. Two methods that are not advanced but found in many production systems are the instant order release and order release by deadline. Orders are released directly as soon as they have been created or once the planned starting date is reached, regardless any production performance measures e.g. the quantity of orders in production [27]. The constant work in process (Conwip) and the load-oriented order release are two common inventory controlling order release approaches. This includes the two heuristics workload-control and bottleneck-control, where

either the workload of all order processing stations or just the stations up to the bottleneck station must be considered. The order release with linear programming differs from the previously presented heuristics in the manner that mathematical optimization is used. A calculation module that can solve linear equations minimizes the target value of an objective function [27]. This is usually used by production software such as ERP or APS, which aim for a direct integration of the order release task into the overall order management process of the organization [4].

3.2 RL-based approaches

Especially in the last two decades, conventional approaches and heuristics have been steadily extended by those based on Artificial Intelligence and here especially the subarea RL [24]. An overview of existing meta-heuristics including learning-based systems is provided in [1]. The reinforcement learning approach introduced by [12] is designed to simultaneously decide on order release and operation sequencing. The Deep Q-Network (DQN) agent tries to minimize the makespan and is evaluated in random simulation instances regarding solution quality, solution speed, and scalability to bigger problems. [28] utilize a deep reinforcement learning approach to determine the release times of the orders in a flow shop with three machines. The reward that the agent receives after every action depends on the number of backorders, current WIP and size of the inventory. The validation results are limited to the simulated case.

Three DQN agents used by [29] independently control three machines to automate scheduling tasks. The model is implemented in MATLAB for training and evaluation purposes and aims to minimize the cycle time spread for three product groups. However, information about the origin of the data is not provided. A centralized learning policy is added to a multiple agent approach by [30]. Individual agents make decisions in a decentralized manner but share a common Q-network. This approach which objective it is to minimize the makespan is validated against 15 generated data sets.

Google DeepMind's AlphaGo Zero algorithm applied to optimize sheet-metal production schedules by [31] interacts with a discrete event simulation and schedules operations to idle machines. The agent aims to jointly minimize tardiness and material waste and is validated using 80 different offline scheduling instances. The multi-step reinforcement learning algorithm introduced by [32] is developed to minimize the total weighted unsatisfied demand in the scheduling horizon. While real industrial datasets are used for evaluation, the validation in this study considers various problem sizes as randomly generated datasets. The approach presented by [33] differs slightly from other reinforcement learning approaches since the DQN agent does not directly decides on the order in this case but selects an operation selection rule and a machine assignment rule. The overall goal to reduce the total tardiness is validated by making assumptions concerning data and by randomly generating test benchmarks. The reinforcement learning mechanism applied in [34] is considered in this paper despite the dispatching focus because of its unique policy transfer. The policy transfer allows it to apply a trained agent in a new factory setting and reduces the effort for model training and data collection. For the validation of the agent aiming to minimize the lateness and tardiness of orders, a simulation based on artificial data is built.

The review on related works supports the three recognized shortcomings of current approaches (see 2.3). First, regarding the problem size it can be seen that when mentioned small instances of 3 to 15 machines are studied and are not usually based on real production data. Using artificial or open-source data sets can help to better fulfil demands on the training data such as data size and independencies but doesn't necessarily support the solution of real-world problems. Second, the mainly used objective is a minimization of the makespan. Only few approaches focus on the adherence to delivery date, customer demand or tardiness. Also regarding action space it is missing, that within action spaces an order is directly chosen for the next scheduling step as a direct consequence from the policy of the algorithm. Lastly, in most cases, no application set-up is described that would facilitate the development and evaluation of algorithms by taking realistic framework conditions on the shop floor with a strong focus on logistic target values into account.

4. Application

This section presents the practice-oriented application approach for an automated order release process, which has been motivated throughout this paper. The main objective aims to optimize adherence to delivery dates. Also, the used online application which allows for a quick validation is introduced.

4.1 Setup of the Application

This approach uses the setup of a RL agent and discrete-event simulation described in our previous paper [1] (see Figure 2). The program architecture of the RL algorithm builds on that of [12] and adapts it in the parts relevant to this paper (see 2.3).

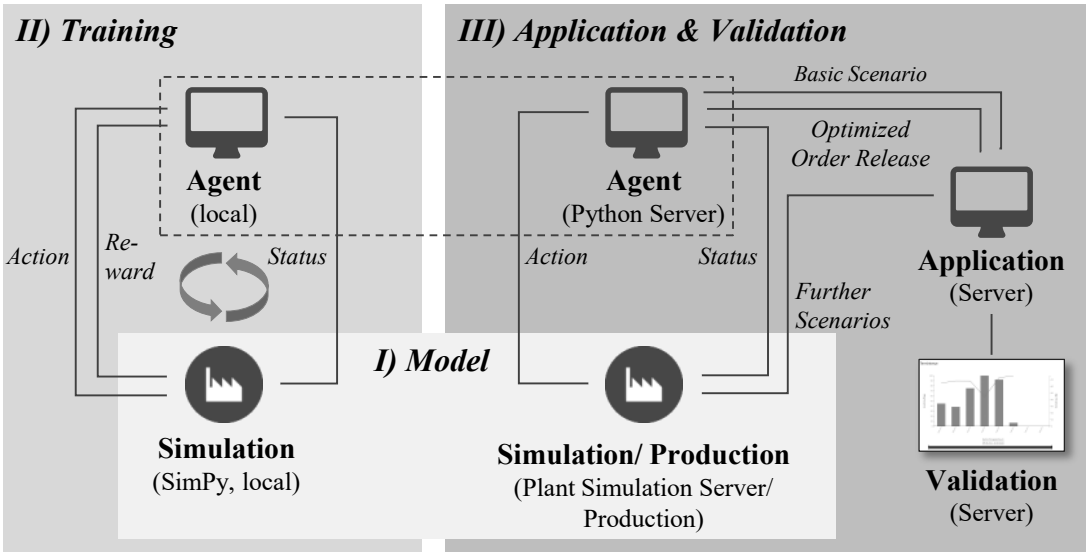

Figure 2: Used setup for the proposed RL approach [1]

In order to model the more-realistic production instance, that is required in this paper, especially the discrete-event simulation in the Plant Simulation environment (see stage III) includes statistically distributed machine breakdowns and order cancellations, the agent must deal with.

4.2 Simulation model and state space

As introduced in the general approach for applying RL algorithms, the current state represented by the state vector provides the agent with all necessary information to decide on his next action. In particular, while some information like work plans and machine lists are loaded once into the simulation environment for its initial creation, data transferred per time step comprises the machine and order status as depicted here:

- **General information:** Current episode, current simulation time
- **Machine status:** Availability, remaining processing time of in-que operations, remaining processing time of the current order
- **Order status:** Machine on which the order is currently processed, downstream machine, processing time of the next process step, remaining processing time of current process step, remaining total lead time of the current order, remaining time until the due date

The general information mainly indicates the simulation progress. In the machine status, for each machine the availability and information on processing times considered with this machine is transferred. Then follow information about each order in production. The vector is successively assembled and its length varies with the number of orders. For calculation of the “remaining total lead time” it is assumed that the waiting times are excluded, so that it is up to the overall system to keep these correspondingly small. By considering the sequence-dependent setup time it is exactly reflected in the calculation of the remaining total lead time.

4.3 Action space

In contrast to [12], in this approach a dependent action space which directly consists of orders to be released or not released at each specific time step is used instead of approximated duration times to which orders must be first assigned. In order to cope with the disadvantage of a dependent action space – its initially defined and constant size – an order release pool has been established which is initially filled with pending orders. The prioritization to fill the order release pool is currently based on the due date of the respective order. In order to be able to link the actions with the corresponding orders in the order pool, each action has an index that matches the index of an order. Once the agent selects an action, the action is used to search for the corresponding order. To depict the case of not releasing any order into the production one more possible action is added into the action space. Besides the possibility of releasing a specific order, the action of not releasing any order is referred to as "No-Op" action.

4.4 Algorithm and reward function

Like the approach of [12], a DQN algorithm learning by experience replay is used with the RL library Stable Baselines. Each step of the agent's experiences is used by the agent in many other steps for weight updates, resulting in a great data efficiency.[35] The biggest change has been done by exchanging a reward on minimizing the makespan to the optimization of adherence to delivery dates (see Table 1). To calculate the reward, the remaining processing time for all orders is determined. An increasing positive and normalized reward is allocated for each order depending on the difference between the remaining time until the due date and the remaining process time. If the difference is negative, the agent receives a negative reward. In addition to this unsteady function, the positive part is multiplied by the value of 10 to enhance positive rewards.

Table 1: Formulation of the algorithm used

Algorithm
Import work plan, list of orders and machine calendar from csv
Initialize state s_0 filled with general information and machine/order status
Define discrete action space
Initialize action space $S \in \{0,1,\dots,M_a\}$ (0,1,…)
Initialize parameters of the DQN library
For episode $e \in \{1,2,\dots,M_e\}$ **do**
Get state vector s_t
Select action from action space
Determine reward r_t
Proceed state to s_{t+1}
End
End

4.5 Control interface and evaluation concept through DAPPS online application

The control interface for rapid evaluation and simple improvement of the RL agent used in this approach is the self-developed online application DAPPS. This tool is based on the approach presented by [36] and has since been further developed and enhanced with more advanced features. DAPPS creates a linkage between the programming and simulation environment and visualizes the agent's order release results by simulating a production scenario. Thus, the agent's decisions on the production environment can be analyzed by comparison against conventional order release scenarios that are simulated as well. By choosing different visualizations of production key figures like adherence to delivery dates, throughput times or Gantt-charts, this helps to quickly derive adjustment possibilities and further develop the algorithms.

The application of the setup presented in this paper, has been conducted on basis of two problem instances: The first with 10 machines and 76 orders, the second with 10 machines and 259 orders. Hypotheses such as the presented objective function or the action space have been derived from expert knowledge and have been directly tested in the DAPPS application with the adherence to delivery dates being the key measure.

As there have been major changes on the problem formulation compared to the first application in [11], the final target of outperform current approaches has not yet been reached. For the smaller data set an adherence to delivery date of 84,21% has already been proven considering maximum freedom the agent got in choosing actions. Then, by applying the second data set with a larger order number, the adherence to delivery dates decreases to 91,89%. Another even larger problem size – with 28 machines and 474 orders – was applied but it emerged that the direct scaling without adjustment in the problem formulation is not purposeful regarding a justifiable training time of around 12 hours. Within the scope of this paper, no solution could be developed yet for the scaling of large problem sizes. Therefore, by using DAPPS as a support system for development, the announced steps (see 2.3.1) must be further carried out here in order to scale the problem.

4.6 Discussion

Within the scope of this paper, we were able to adjust the action space so that the action space is dependent and directly consist of orders to be released. To deal with a dependent action space, we additional add an order release pool. In addition, we are introducing the DAPPS tool to enable an evaluation of release results and a comparison against conventional heuristics such as CONWIP. For the application we use two scenarios with different numbers of orders and 10 machines each. A larger problem size with more machines and 474 orders led to a long training period. The problem of scaling could not be solved within the scope of the paper.

5. Conclusion and further research

This paper elaborates on the usage of reinforcement learning algorithms for automated order release in a practice-based application. Therefore it aims to further develop the reinforcement learning agent presented in our previous paper and introduces the evaluation tool DAPPS. After identifying the shortcomings of current RL approaches, the functionality of our agent and the advancements compared to the previous agent has been explained. Finally, by embedding this practice-based approach into DAPPS, the tool has been presented and the agent been tested on two problem sizes.

The approach can be used in practice, in order release planning to support decision making and thus lead to a better achievement of logistical target values as well as to a reduction of ad-hoc control interventions. The focus of further research must be on an improvement of the problem formulation and on the identification of the computationally intensive component. Finally, the approach has to be validated and the behaviour of the algorithm’s decision needs to be evaluated by means of unforeseen disruptions in the production.

Acknowledgements

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

References

[1] Schuh, G., Gützlaff, A., Schmidhuber, M., Fulterer, J., Janke, T., 2022. Towards an Automated Application for Order Release. Procedia CIRP 107, 1323–1328.

[2] Schlegel, T., Siegert, J., Bauernhansl, T., 2019. Metrological Production Control for Ultra-flexible Factories. Procedia CIRP 81, 1313–1318.

[3] Haeussler, S., Stampfer, C., Missbauer, H., 2020. Comparison of two optimization based order release models with fixed and variable lead times. International Journal of Production Economics 227, 107682.

[4] Schuh, G., Stich, V., Reuter, C., Blum, M., Brambring, F., Hempel, T., Reschke, J., Schiemann, D., 2018. Cyber Physical Production Control, in: Jeschke, S., Brecher, C., Song, H., Rawat, D.B. (Eds.), Industrial Internet of Things. Cybermanufacturing Systems. Springer, Cham, pp. 519–539.

[5] Lang, S., Schenk, M., Reggelin, T., 2019. 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 52 (13), 2716–2721.

[6] Pürgstaller, P., Missbauer, H., 2012. Rule-based vs. optimisation-based order release in workload control: A simulation study of a MTO manufacturer. International Journal of Production Economics 140 (2), 670–680.

[7] Maetschke, J., Fulterer, J., Janke, T., Zipfel, A., Bank, L., Theumer, P., Mundt, C., Köster, N., Kämpfer, T., Heuer, T., Hiller, T., 2022. PPS-Report 2021. Zeitschrift für wirtschaftlichen Fabrikbetrieb 117 (6), 400–404.

[8] Gutenschwager, K., Rabe, M., Spieckermann, S., Wenzel, S., 2017. Simulation in Produktion und Logistik: Grundlagen und Anwendungen. Springer Vieweg, Berlin, 281 pp.

[9] Panzer, M., Bender, B., Gronau, N., 2021. Deep Reinforcement Learning In Production Planning And Control: A Systematic Literature Review.

[10] Zeng, D., Gu, L., Pan, S., Cai, J., Guo, S., 2019. Resource Management at the Network Edge: A Deep Reinforcement Learning Approach. IEEE Network 33 (3), 26–33.

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

[12] Samsonov, V., Kemmerling, M., Paegert, M., Lütticke, D., Sauermann, F., Gützlaff, A., Schuh, G., Meisen, T., 2021. 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.

[13] Roderick, L.M., Phillips, D.T., Hogg, G.L., 1992. A comparison of order release strategies in production control systems. International Journal of Production Research 30 (3), 611–626.

[14] Vinod, V., Sridharan, R., 2011. Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. International Journal of Production Economics 129 (1), 127–146.

[15] Gonçalves, J.F., Magalhães Mendes, J.J. de, Resende, M.G., 2005. A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research 167 (1), 77–95.

[16] Xie, J., Gao, L., Peng, K., Li, X., Li, H., 2019. Review on flexible job shop scheduling. IET Collaborative Intelligent Manufacturing 1 (3), 67–77.

[17] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

[18] Hubbs, C.D., Li, C., Sahinidis, N.V., Grossmann, I.E., Wassick, J.M., 2020. A deep reinforcement learning approach for chemical production scheduling. Computers & Chemical Engineering 141, 1–22.

[19] Waschneck, B., 2020. Autonome Entscheidungsfindung in der Produktionssteuerung komplexer Werkstattfertigungen. Dissertation.

[20] Sutton, R.S., Barto, A., 2018. Reinforcement learning, second edition: An introduction, Second edition ed. The MIT Press, Cambridge, Massachusetts, London, England, 590 pp.

[21] Overbeck, L., Hugues, A., May, M.C., Kuhnle, A., Lanza, G., 2021. Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. Procedia CIRP 103, 170–175.

[22] Shannon, R.E., 1998. Introduction to the art and science of simulation, in: Proceedings of the 1998 Winter Simulation Conference. Winter Simulation Conference, Washington, DC. IEEE, pp. 7–14.

[23] Chen, M., Mao, S., Liu, Y., 2014. Big Data: A Survey. Mobile Netw Appl 19 (2), 171–209.

[24] Kayhan, B.M., Yildiz, G., 2021. Reinforcement learning applications to machine scheduling problems: a comprehensive literature review. J Intell Manuf.

[25] Abdi, A.H., Abolmaesumi, P., Fels, S., 2019. Variational Learning with Disentanglement-PyTorch. Journal of Machine Learning Research 1:1-6.

[26] Luus, F., Khan, N., Akhalwaya, I., 2019. Active Learning with TensorBoard Projector.

[27] Lödding, H., 2016. Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration. Springer Berlin Heidelberg, Berlin, Heidelberg, 679 pp.

[28] Schneckenreither, M., Haeussler, S., 2019. Reinforcement Learning Methods for Operations Research Applications: The Order Release Problem, in: Nicosia, G., Pardalos, P.M., Giuffrida, G., Umeton, R., Sciacca, V. (Eds.), Machine Learning, Optimization and Data Science. Springer International Publishing, pp. 545–559.

[29] Waschneck, B., Reichstaller, A., Belzner, L., Altenmuller, T., Bauernhansl, T., Knapp, A., Kyek, A., 2018. Deep reinforcement learning for semiconductor production scheduling, in: 2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA. 30.04.2018 - 03.05.2018. IEEE, pp. 301–306.

[30] Park, I.-B., Huh, J., Kim, J., Park, J., 2020. A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities. IEEE Trans. Automat. Sci. Eng. 17 (3), 1420–1431.

[31] Rinciog, A., Mieth, C., Scheikl, P.M., Meyer, A., 2020. Sheet-Metal Production Scheduling Using AlphaGo Zero, in: Nyhuis, P., Herberger, D., Hübner, M. (Eds.), Proceedings of the Conference on Production Systems and Logistics, pp. 342–352.

[32] Zhang, Z., Zheng, L., Hou, F., Li, N., 2011. Semiconductor final test scheduling with Sarsa(λ,k) algorithm. European Journal of Operational Research 215 (2), 446–458.

[33] Luo, S., 2020. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. Applied Soft Computing 91.

[34] Zheng, S., Gupta, C., Serita, S., 2019. Manufacturing Dispatching using Reinforcement and Transfer Learning, in: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (Eds.), Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 655–671.

[35] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing Atari with Deep Reinforcement Learning, 9 pp.

[36] Schuh, G., Potente, T., Fuchs, S., Schmitz, S., 2012. Wertstromorientierte Produktionssteuerung: Interaktive Visualisierung durch IT-Tools zur Bewertung der Logistik- und Produktionsleistung. wt Werkstattstechnik online 102 (4), 176–180.

Biography

Günther Schuh (*1958) holds the Chair of Production Systems at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University, is a member of the board of directors of the Fraunhofer Institute for Production Technology IPT and director of the Research Institute for Rationalization (FIR) at the RWTH Aachen.

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 at RWTH Aachen University.

Jan Maetschke (*1994) studied Mechanical Engineering (Production Engineering) at RWTH Aachen University in Germany. He 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.

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 at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University and Team Lead Software Development in the department.

Hendrik Eisbein (*1998) studied Business Administration and Engineering (Mechanical Engineering) at the RWTH Aachen University in Germany. He is a Student Research Assistant in Production Logistics at the Laboratory for Machine Tools and Production Engineering WZL at RWTH Aachen University.