Simulation in Produktion
und Logistik 2021
Jörg Franke & Peter Schuderer (Hrsg.)
Cuvillier Verlag, Göttingen 2021

Scalable cooperative multi-agent-reinforcement-learning for order-controlled on schedule manufacturing in flexible manufacturing systems

Skalierbares kooperatives Multi-Agent-Reinforcement-Learning zur termintreuen, auftragsgesteuerten Fertigungssteuerung in flexiblen Fertigungssystemen

Berend Denkena, Marc-André Dittrich, Silas Fohlmeister,
Daniel Kemp, Gregory Palmer
Leibniz University Hannover, Hannover (Germany),
denkena@ifw.uni-hannover.de, dittrich@ifw.uni-hannover.de,
fohlmeister@ifw.uni-hannover.de, kemp@ifw.uni-hannover.de, gpalmer@l3s.de

Abstract: To operate flexible manufacturing systems efficiently, a robust and optimal production control is crucial. With an increasing number of workpieces being processed in parallel, ensuring guaranteed lead times represents a complex optimization tasks, better known as the *flexible scheduling problem*. Cooperative multi-agent reinforcement learning approaches have recently shown their potential in production control. However, ensuring guaranteed lead times in flexible manufacturing systems with these approaches remains an open problem. In this work, an existing cooperative multi-agent framework for flexible job-shop scheduling is transferred and modified to optimize production control in flexible manufacturing systems. Using a centralized training for decentralized execution multi-agent deep reinforcement learning approach, the goal is to optimize order agents to ensure guaranteed lead times. Furthermore, a comprehensive simulation study investigates the effect of common knowledge on facilitating cooperation, and empirically evaluate the frameworks scalability to a range of challenging scenarios.

1 Introduction

The trend towards an increasing customization of workpieces in the manufacturing industry, while ensuring guaranteed lead times, can be facilitated with flexible manufacturing systems (FMS) (ElMaraghy et al., 2013). FMS typically consist of a group of universal CNC-machine tools, which are interconnected by an automated material handling and storage system for workpieces and cutting tools (Brecher and Weck, 2019). Furthermore, FMS are automatically setup and loaded and thus are able

to attain high flexibility and high productivity without any manual interventions (Manu et al., 2018). However, with an increasing number and range of machine tools and the number of workpieces processed in parallel, a robust production control is crucial for operating a FMS efficiently (Kutin et al., 2018). The allocation of workpieces to dedicated machine tools, e. g. for ensuring granted lead times, is critical, representing a complex optimization tasks, known as the flexible scheduling problem (Sormaz and Patel, 2018).

Cooperative multi-agent systems (MAS) represent a promising approach to achieve a robust and optimized production control in fully automated FMS (Monostori et al., 2016). Their decentralized and cooperative way of solving optimization problems, enable them to cope with complex control tasks (Leitão, 2009). Due to these characteristics MAS are essential for realising an order-controlled production, in which workpieces autonomously coordinate themselves through a manufacturing environment, e.g. a FMS (Bussmann et al., 2004). To enable the intelligent control and steady optimization, deep reinforcement learning (RL) techniques were recently applied to MAS, e.g. (Waschneck et al., 2018b). A major constraint of RL is the dependence on simulation environments (Gallina et al., 2019).

Dittrich and Fohlmeister (2020) proposed a cooperative multi-agent reinforcement learning (MARL) framework, which is integrated in *Tecnomatix Plant Simulation*. Therein, intelligent order-agents are able to solve flexible job-shop scheduling problems efficiently, through learning to cooperate. The authors have shown the approach's potential for successfully reducing mean cycle times in case of production control for job shop production environments. Their approach can also be transferred to the related production control domains, which can be characterized a *flexible scheduling problem* as well. Thus, this article will modify the aforementioned framework for ensuring guaranteed lead times in FMS through expanding the so-called *field-of-view common knowledge*, i. e., the number of agents sharing information between each other. Schroeder de Witt et al. (2018) observed that a large number of cooperative multi-agent tasks benefit from considering it, as its size can increase the likelihood of agents learning to coordinate. To the best of our knowledge, to date no analysis has been performed regarding the impact of field-of-view-size on the performance of MARL agents learning to solve the flexible scheduling problem.

2 State of the Art

As mentioned in the previous section, deep RL-approaches were successfully applied to flexible scheduling problems. Similar approaches mainly differ regarding the underlying optimization goal and their optimization approach. Thus, similar optimization approaches are briefly categorized in Table 1.

***Table 1**: Key characteristics of similar approaches*

Reference	Goal of optimization	Optimization approach
(Dittrich and Fohlmeister, 2020)	Mean cycle times	Cooperative multi-agent approach (CMAA)
(Göppert et al., 2020)	Capacity utilization	Single-agent approach

(Guo et al., 2020)	Tardiness	Single-agent approach
(Han and Yang, 2020)	Total production time	Single-agent approach
(Hofmann et al., 2020)	Run-trough time	Multi-agent approach
(Hu et al., 2020)	Avoidance of downtimes	Multi-agent approach
(Kim et al., 2020)	Tardiness	CMAA*
(Liu et al., 2020)	Total production time	Single-agent approach
(Luo, 2020)	Tardiness	Single-agent approach
(Shiue et al., 2020)	Run-through time	Single-agent approach
(Zhu et al., 2020)	Total production time	Single-agent approach
(Baer et al., 2019)	n. a.	Multi-agent approach
(Qu et al., 2019)	Work-in-Progress	Multi-agent approach
(Silva and Azevedo, 2019)	Lead times	Single-agent approach
(Silva et al., 2019)	Total production time	CMAA
(Qu et al., 2018)	Total production costs	Multi-agent approach
(Wang et al., 2018)	Run-trough time and total production costs	Multi-agent approach
(Waschneck et al., 2018a)	Capacity utilization	CMAA
(Waschneck et al., 2018b)	Run-through time and cycle times	Single-agent approach
(Bouazza et al., 2017)	Waiting time	Multi-agent approach
(Shahrabi et al., 2017)	Total production costs	Single-agent approach
(Qu, 2016)	Total production costs	CMAA
(Qu et al., 2016)	Total production costs	Multi-agent approach

* Scheduling via intelligent machine agents

It can be seen that only few proposed approaches exist, which focus on optimizing the tardiness, i. e., ensuring guaranteed lead times. In addition, most of the approaches do not make any use of common knowledge between agents. None of them were applied to tardiness-optimization in this context. In case cooperative agents were used, no author has comprehensively analysed the size of the field-of-view's effect on the quality of resulting joint-policy, i. e., the extent to which it facilitates cooperation between intelligent agents. However, as stated in Schroeder de Witt et al. (2018), an increased field-of-view size, and thereby increased common knowledge, enables complex decentralised coordination. Therefore, the field-of-view size should be considered.

3 Formulation of the optimization problem

Guarantying lead times or in other words preventing tardiness, is highly relevant in context of production control. The tardiness T is in our case defined as the percentage difference between a guaranteed lead time and the time of completion of an individual workpiece. As any tardiness T is in general unfavourable, the goal of optimization should however not be confused with just minimizing the overall tardiness. A plain minimisation would lead to increasing stock of finished parts and also to increasing

capital commitment. Instead, our goal is to reach an overall mean tardiness $\overline{T}$ of ideally zero percent or slightly below zero percent, while controlling a dynamically generated production program.

Since the underlying flexible scheduling problems can be described as a fully-cooperative multi-agent task, it can be formulated as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek and Amato, 2016). The Dec-POMDP is a tuple (*n, X, O, U, P, R*) consisting of a state space *X*, an observation function (see Eq. 1), for each state $x \in X$ a joint action space $U \equiv U^n$, a transition function (see Eq. 2) returning the probability of transitioning from a state x_t to x_{t+1} given an action profile u, and a reward function (see Eq. 3) for each agent *a*. The agents are optimized using local and global rewards. The global reward function is shared. Finally, we allow *terminal* (absorbing) states at an episode's end.

$$O_i: X \to \mathbb{R}^d \tag{1}$$

$$P: X_t \times U \times X_{t+1} \to [0,1] \tag{2}$$

$$R_i: X_t \times U \times X_{t+1} \to R \tag{3}$$

The presented MARL-framework applies a centralized training for decentralized execution approach using deep Q-learning (DQN) (Mnih et al., 2015). Agents share a multi-layer perceptron, trained to approximate Q-Values for observation-action pairs (see Eq. 4). Network parameters θ are trained using Adam (Kingma and Ba, 2015) on the mean squared Bellman residual with the expectation taken over transitions uniformly sampled from an experience replay memory (see Eq. 5), where Y_t is the defined target (see Eq. 6). Parameters θ_t' belong to a stable *target network*, which is synchronised with the current network every *n* transitions (Mnih et al., 2015).

$$Q_i: O_i \times A_i \to \mathbb{R} \tag{4}$$

$$L(\theta i) = \mathbf{E}_{o,a\sim p(\cdot)}[(Yt - Q(o,a;\theta t))^2] \tag{5}$$

$$Y_t \equiv r_{t+1} + \gamma \operatorname{argmax}_{\alpha\in A} Q(o_{t+1}, \alpha; \theta_t') \tag{6}$$

4 Modifications and hyperparameter settings

The MARL framework (Fig. 1) contains a central DQN module and a decentralized cooperative MAS. Via socket-interfaces the communication to the simulation software *Tecnomatix Plant Simulation* is realised, in which an FMS can be modelled. Workpieces to be processed in the simulated FMS are represented as generic order agents a_i in the cooperative MAS, available machine tools are represented as generic machine agents m_i, respectively. Relevant machine agents $m_{i,t}$ are requested by individual order agents a_i to process the equivalent workpieces. They will propose the request with an expected timeslot for processing. In case no processing is possible, the machine agent $m_{i,t}$ will decline the request. In order to also consider other order agents $\{a_{i+1}, \ldots, a_n\}$ in the decision making process, a group of cooperating agents can be requested to provide information about themselves and their current production process. An agent's field-of-view is therefore determined by the number of agents

providing information. After all relevant information is received by the order agent a_i, the information is aggregated and a decision is made through evaluating the latest available local function approximator Q_i. A more detailed description is provided in Dittrich and Fohlmeister (2020).

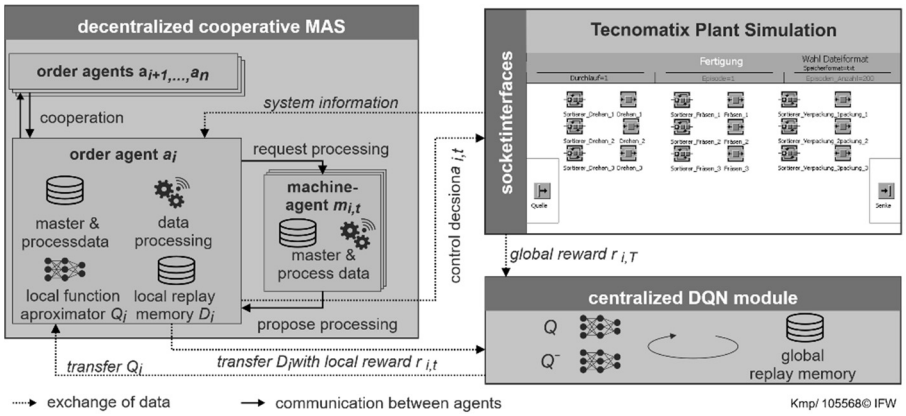


Figure 1: *MARL-Framework by Dittrich and Fohlmeister (2020)*

The outlined decision-making process can be fully adopted for production control of various FMS. A few modifications have to be considered for ensuring that the DQN module trains the local function approximator Q_i towards an optimal mean tardiness $\bar{T}$. First and foremost, the global reward function $r_{i,T}$ needs to be modified. For training the network parameters of Q_i regarding an optimal tardiness, preliminary investigations have shown that good results can be achieved with a $r_{i,T}$ according to Equation 7:

$$r_{i,T} = \begin{cases} 0 & \Delta_i < -20\% \\ 3 & -20\% \leq \Delta_i < -10\% \\ 10 & -10\% \leq \Delta_i \leq 0\% \\ -5 & 0\% < \Delta_i. \end{cases} \tag{7}$$

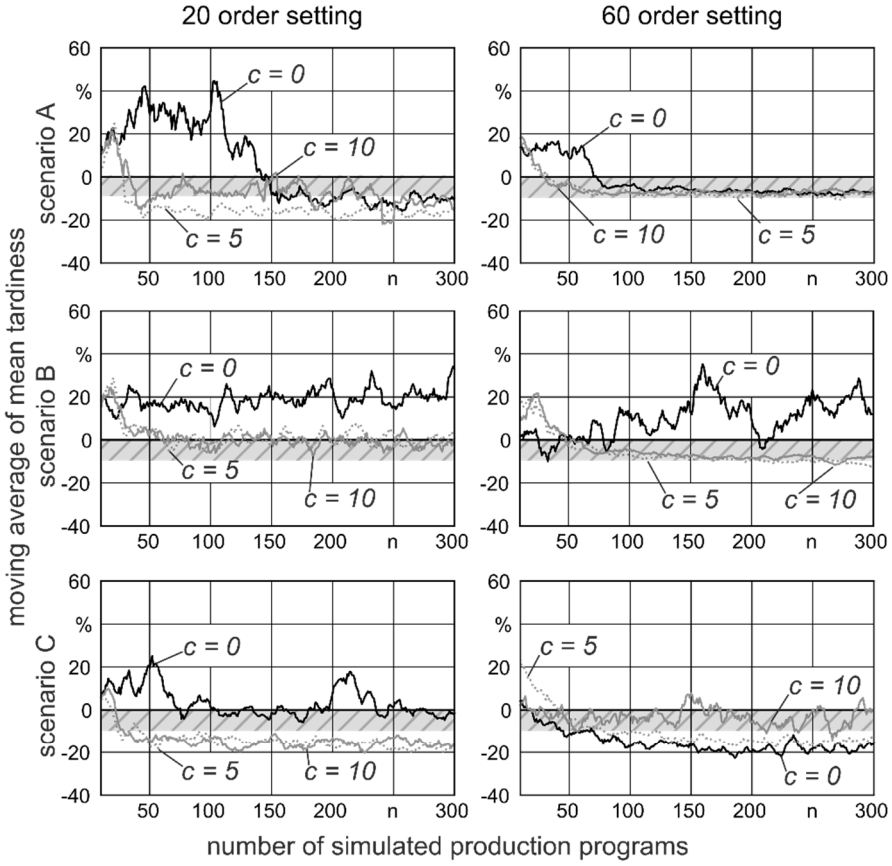
The combination of local and global rewards leads to a comparatively fast convergence towards a nearly optimal tardiness. In comparison to the fundamental framework the use of a larger batch size in the DQN module was implemented. Considering the performance of the simulation studies in section 5, the hyperparameter setting of Table 2 were evaluated as working well by the authors.

Table 2: *Hyperparamter settings of the DQN-approach*

Hyperparameter settings	
Warmup steps n_{warmup}:	3,000
Capacity D of global replay memory:	20,000
Exploration rate ε, decay ε_d of ε, ε_{min}:	1.2, 0.9999, 0.001
Learning rate α, discount factor γ, batch size:	0.001, 0.95, 50

5 Simulation Study

To analyse the effect of an increased common knowledge between cooperating agents, a comprehensive simulation study has been executed. To empirically evaluate the scalability of the outlined approach, a total of eighteen configurations has been composed (Fig 2). Therefore, three FMS were designed and gradually scaled. First, an FMS with three manufacturing processes (MP) and three machines per MP has been modelled (Scenario A). Second, the FMS was scaled by duplicating the number of machine tools per MP (Scenario B). The production program consisted of three, dynamically and randomly created order types.



Details of the flexible manufacturing system

	Scenario A	Scenario B	Scenario C
Number of processes *p*:	3	3	5
Machine tools per process *m*:	3	6	6
Order types in production program *t*:	3	3	6
Size of neural network layers:	8x8x8	20x10x6	70x35x12

Kmp/ 105576 © IFW

Figure 2: *Results of the simulation study concerning the common knowledge effect*

Third, the number of MP was increased to a total of five, while leaving the number of machine tools per MP constant (scenario C). The production program's complexity was scaled to six order types. All scenarios were simulated for production programs consisting of twenty as well as sixty orders. In each scenario the field-of-view's size *c* has been varied between zero, five or ten consecutive order agents, which were included into the common knowledge of the decision-making order agent. The framework and all outlined scenarios are provided in detail in Palmer et al. (2021).

Considering the results for scenario A, it can be seen that the MARL approach was able to successfully learn a control-policy in all simulated configurations, which at least ensured a mean tardiness $\overline{T}$ of zero percent or below. Thus, any tardiness was successfully avoided and guaranteed lead times were satisfied. However, without any cooperating agents (*c=0*) it took significantly longer to reach the desired tardiness corridor (hatched area). Each increase of the field-of-view size resulted in an improved performance. As can be observed in the sixty-order-setting, in which the number of executed control decisions tripled compared to the twenty-order-setting, all simulated scenarios reached a similar as well as stable control-policy in the long run.

Scaling up the FMS (scenario B), i. e., increasing the underlying complexity of the *flexible scheduling problem,* lead to a significantly different performance of the *c=0* configurations. Without any common knowledge the *c=0* control policies did not converge successfully, meaning that the desired tardiness corridor was never reached. The other field-of-view-configurations were able to reach the desired corridor, with *c=10* slightly outperforming *c=5*. The sixty-order-setting, lead to an even more stable performance. Yet, due to the higher complexity, these configurations were only able to reach a near optimal $\overline{T}$ of about negative ten percent in the long run.

Considering scenario C, *c=5* and *c=10* again outperformed the *c=0* configuration. In the twenty-order-setting *c=0* was at no time able to stably avoid positive $\overline{T}$. For the most complex sixty-order-setting, ten verification runs were conducted. Therefore, average results are presented. It can be observed that *c=0* reached a reasonable performance regarding its performance average. This could be explained due to no common knowledge of *c=0*, resulting in achieving a fast run-through time triggered by the use of local rewards. Despite this fact, it was not able to reach the desired tardiness corridor and was outperformed by *c=5* and *c=10*, as well. Hence, the general observation can be derived that considering common knowledge positively affects the approach's performance.

6 Conclusion and Future Outlook

Within FMS guaranteeing lead times, while avoiding tardiness and reducing capital commitment in parallel, represents a complex optimization task. Cooperative MARL appears as a promising solution. However, the factor of common knowledge between cooperating agents is crucial. To evaluate the effect of an increased common knowledge in context of scalability and performance, a comprehensive simulation study has been executed. The study has shown that the *flexible scheduling problem* becomes significantly less manageable without any consideration of cooperating agents. The consideration leads to a faster convergence towards an optimal control policy and results in a better performance. Nevertheless, its exact size needs to be critically evaluated case-by-case, as a broader common knowledge only slightly improved the performance, while increasing the computational complexity. Future

research should therefore consider further simulation studies for determining a rule of thumb effect of common knowledge on multi-agent learning in context of production control. In addition, the presented approach only includes workpieces and machinery so far. Thus, future research should also consider tangential processes, i. e., material and tool supply, as those are of high relevance for actual applications in practice.

Acknowledgements

The authors gratefully acknowledge, that the proposed research is a result of the research project “IIP-Ecosphere”, granted by the German Federal Ministry of Economics and Technology (BMWi) via funding code 01MK20006A.

References

Baer, S.; Bakakeu, J.; Meyes, R.; Meisen, T.: Multi-Agent Reinforcement Learning for Job Shop Scheduling in Flexible Manufacturing Systems. In: 2nd International Conference on Artificial Intelligence for Industries (AI4I), Laguna Hills, CA, USA, 2019, pp. 22–25.

Bouazza, W.; Sallez, Y.; Beldjilali, B.: A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect. IFAC-PapersOnLine 50 (2017) 1, pp. 15890–15895.

Brecher, C.; Weck, M.: Werkzeugmaschinen Fertigungssysteme 1. Berlin, Heidelberg: Springer 2019.

Bussmann, S.; Jennings, N.R.; Wooldridge, M.: Multiagent Systems for Manufacturing Control. Berlin, Heidelberg: Springer Berlin Heidelberg 2004.

Dittrich, M.-A.; Fohlmeister, S.: Cooperative multi-agent system for production control using reinforcement learning. CIRP Annals 69 (2020) 1, pp. 389–392.

ElMaraghy, H.; Schuh, G.; ElMaraghy, W.; Piller, F.; Schönsleben, P.; Tseng, M.; Bernard, A.: Product variety management. CIRP Annals 62 (2013) 2, pp. 629–652.

Gallina, V.; Lingitz, L.; Karner, M.: A New Perspective of the Cyber-Physical Production Planning System. In: 16th IMEKO TC10 Conference, Berlin, Germany, 2019, 2019, pp. 60–65.

Göppert, A.; Rachner, J.; Schmitt, R.H.: Automated scenario analysis of reinforcement learning controlled line-less assembly systems. Procedia CIRP 93 (2020), pp. 1091–1096.

Guo, L.; Zhuang, Z.; Huang, Z.; Qin, W.: optimization of dynamic multi-objective non-identical parallel machine scheduling with multi-stage reinforcement learning. In: IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, Hong Kong, 2020, pp. 1215–1219.

Han, B.-A.; Yang, J.-J.: Research on Adaptive Job Shop Scheduling Problems Based on Dueling Double DQN. IEEE Access 8 (2020), pp. 186474–186495.

Hofmann, C.; Krahe, C.; Stricker, N.; Lanza, G.: Autonomous production control for matrix production based on deep Q-learning. Procedia CIRP 88 (2020), pp. 25–30.

Hu, L.; Liu, Z.; Hu, W.; Wang, Y.; Tan, J.; Wu, F.: Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. Journal of Manufacturing Systems 55 (2020), pp. 1–14.

Kim, Y.G.; Lee, S.; Son, J.; Bae, H.; Chung, B.D.: Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. Journal of Manufacturing Systems 57 (2020), pp. 440–450.

Kingma, D.P.; Ba, J.: Adam: A Method for Stochastic Optimization. In: 3rd International Conference for Learning Representations, San Diego, CA, USA, 2015, pp. 1–15.

Kutin, A.A.; Dolgov, V.A.; Kabanov, A.A.; Dazuk, I.V.; Podkidyshev, A.A.: Improving the efficiency of CNC machine tools with multi-pallet systems in machine-building manufacturing. IOP Conference Series: Materials Science and Engineering 448 (2018), pp. 12010.

Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence 22 (2009) 7, pp. 979–991.

Liu, C.-L.; Chang, C.-C.; Tseng, C.-J.: Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. IEEE Access 8 (2020), pp. 71752–71762.

Luo, S.: Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. Applied Soft Computing 91 (2020), pp. 106208.

Manu, G.; Vijay Kumar, M.; Nagesh, H.; Jagadeesh, D.; Gowtham, M.B.: Flexible Manufacturing Systems (FMS): A Review. International Journal of Mechanical and Production Engineering Research and Development (IJMPERD) 8 (2018) 2, pp. 323–336.

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.: Human-level control through deep reinforcement learning. Nature 518 (2015) 7540, pp. 529–533.

Monostori, L.; Kádár, B.; Bauernhansl, T.; Kondoh, S.; Kumara, S.; Reinhart, G.; Sauer, O.; Schuh, G.; Sihn, W.; Ueda, K.: Cyber-physical systems in manufacturing. CIRP Annals 65 (2016) 2, pp. 621–641.

Oliehoek, F.A.; Amato, C.: A Concise Introduction to Decentralized POMDPs. Cham: Springer International Publishing 2016.

Palmer, G.; Kemp, D.; Fohlmeister, S., 2021: Scalable cooperative Multi-Agent-Reinforcement-Learning for order-controlled on schedule manufacturing in flexible manufacturing systems. https://github.com/gjp1203/fms_marl, accessed May 14th, 2021.

Qu, S.: Learning Adaptive Dispatching Rules for a Manufacturing Process System by Using Reinforcement Learning Approach. In: 21st International Conference on Emerging Technologies and Factory Automation (EFTA), Berlin, Germany, 2016, pp. 1–8.

Qu, S.; Wang, J.; Govil, S.; Leckie, J.O.: Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-skill Workforce and Multiple Machine Types: An Ontology-based, Multi-agent Reinforcement Learning Approach. Procedia CIRP 57 (2016), pp. 55–60.

Qu, S.; Wang, J.; Jasperneite, J.: Dynamic scheduling in large-scale stochastic processing networks for demand-driven manufacturing using distributed reinforcement learning. In: IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 2018, pp. 433–440.

Qu, S.; Wang, J.; Jasperneite, J.: Dynamic scheduling in modern processing systems using expert-guided distributed reinforcement learning. In: IEE 24th International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 2019, pp. 459–466.

Schroeder de Witt, C.A.; Foerster, J.N.; Farquhar, G.; Torr, P.H.; Boehmer, W.; Whiteson, S.: Multi-Agent Common Knowledge Reinforcement Learning. In: 32nd Conference on Neural Information Processing Systems (NeurlPS), Montréal, Canada, 2018, pp. 1–17.

Shahrabi, J.; Adibi, M.A.; Mahootchi, M.: A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. Computers & Industrial Engineering 110 (2017), pp. 75–82.

Shiue, Y.-R.; Lee, K.-C.; Su, C.-T.: A Reinforcement Learning Approach to Dynamic Scheduling in a Product-Mix Flexibility Environment. IEEE Access 8 (2020), pp. 106542–106553.

Silva, M.A.; Souza, S.R. de; Freitas Souza, M.J.; Bazzan, A.L.: A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. Expert Systems with Applications 131 (2019), pp. 148–171.

Silva, T.; Azevedo, A.: Production flow control through the use of reinforcement learning. Procedia Manufacturing 38 (2019), pp. 194–202.

Sormaz, D.; Patel, C.: Development and evaluation of feature-focused dynamic routing policy. The International Journal of Advanced Manufacturing Technology 99 (2018) 1-4, pp. 15–28.

Wang, M.; Chen, X.; Zhou, J.; Jiang, T.; Cai, W.: Shared Cognition Based Integration Dynamic Scheduling Method. In: IEEE 2nd Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC), Xi'an, China, 2018, pp. 1438–1442.

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

Waschneck, B.; Reichstaller, A.; Belzner, L.; Altenmüller, T.; Bauernhansl, T.; Knapp, A.; Kyek, A.: Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP 72 (2018b), pp. 1264–1269.

Zhu, J.; Wang, H.; Zhang, T.: A Deep Reinforcement Learning Approach to the Flexible Flowshop Scheduling Problem with Makespan Minimization. In: IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), Liuzhou, China, 2020, pp. 1220–1225.