Technology Readiness Levels of Reinforcement Learning methods for simulation-based production scheduling

Arne Seipolt[1], Ralf Buschermöhle[1], Maximilian Höfinghoff[1], Goy-Hinrich Korn[2], Marcel Schumacher[1]

Abstract: Digital Twins (DT) are nowadays widely used and provide a benefit for the companies using it. One service of the DT is the simulation of a production process. This enables an optimization of the production process by simulation optimization, for example with Reinforcement Learning (RL). To support researchers and practitioners in deciding which algorithm is suitable for an implementation under real-life conditions, a literature research is performed, and a Machine Learning Technology Readiness Level is assigned to the different RL-Algorithms. It can be shown that recent research focuses mainly on model free value based and evolutionary algorithms, and both are suitable for an implementation in a real-world scenario. Both algorithms can outperform widely applied dispatching rules. Nevertheless, it should be evaluated why other algorithms are not in the focus of recent research and how the algorithms perform in comparison to each other.

Keywords: Technology Readiness, Reinforcement Learning, Simulation, Production Scheduling

1 Introduction

Digital Twins (DT) provide a benefit for the companies using it. A study from 2022 shows, that the companies, which are using Digital Twins, could improve their operational efficiency on average by 15 % [Gy22]. Furthermore, 68 % of the asked organizations see a simulation as a service from the DT, justifying its use [Gy22]. For example, it is possible to optimize the production scheduling by using a simulation [Sc07]. But there is still potential for further development [Mo20]. While there are different methods to optimize the parameters of a simulation [LTD22], these optimizations can often only be done approximately [La17].

Recently, Reinforcement Learning (RL) methods have achieved great successes in playing different games and are able to outperform human players [Mn15]. These algorithms can be used for simulation optimization by suggesting promising simulation parameters [PBG22, KD21]. In a literature research, Panzer et al. reviewed a total of 55 recent scientific articles that deal with production scheduling using reinforcement learning. However, 52 of these were implemented and validated exclusively in a laboratory

[1] Osnabrück University of Applied Sciences, Faculty of Management, Culture and Technology, Kaiserstraße 10c, D-49809 Lingen/Ems, {a.seipolt, r.buschermoehle, m.hoefinghoff, marcel.schumacher}@hs-osnabrueck.de

[2] Bernard Krone Holding GmbH & Co. KG, CIO & CDO, 48480 Spelle, Heinrich-Krone-Straße 10, goy-hinrich.korn@krone.de

environment. Therefore, no general statements can be made about the reliability of such systems in reality [PBG22].

To support researchers and practitioners in deciding which method is suitable for an implementation under real-life conditions, this paper aims to show different reinforcement learning methods for simulation-based optimization of production processes. In order to show the technology readiness of these methods, they will be assigned a "Machine Learning Technology Readiness Level" (MLTRL), defined in [La22].

For this purpose, a definition of production scheduling and an overview of different reinforcement learning algorithms is given, followed by an introduction to the MLTRL. Then a literature research is conducted, considering only papers published in recent years that deal with reinforcement learning for simulation optimization in the context of production planning. These publications are classified based on the algorithms used, in order to check to which MLTRL the implementations and evaluations performed correspond. This is followed by a discussion to derive further research approaches and implementation strategies.

2 Production Scheduling

"Scheduling problems can be understood in general as the problems of allocating resources over time to perform a set of tasks being parts of some processes, among which computational and manufacturing ones are most important." [Bl19] "In manufacturing, the purpose of production scheduling is to minimize production time and costs, by telling a production facility when to make something, with which staff, and using which equipment." [Ri12] Scheduling problems can be characterized by three sets: $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$ defines the different tasks, $\mathcal{P} = \{P_1, P_2, \dots, P_m\}$ defines the processors or machines and $\mathcal{R} = \{R_1, R_2, \dots, R_s\}$ defines additional resources. Different tasks can be combined to jobs. So, job J_j is divided into n_j tasks: $T_{1j}, T_{2j}, \dots, T_{n_j j}$ while different tasks are performed on different machines. Under the condition, that every machine is specialized for the execution of certain tasks, there are three models for processing: *flow shop*, *open shop* and *job shop*. In an open shop, the number of Tasks is equal for every job, and the task T_{ij} is performed on the machine P_i. Additionally, in flow shop, the processing of T_{i-1j} precedes the processing of T_{ij}. In a job shop, n_j is arbitrary [Bl19].

To introduce the complexity of Production Scheduling, the number of possible solutions for a simple Production Scheduling problem will be reviewed. It is assumed, that n different tasks should be performed to finish a Job. The number of tasks per job is constant. For every task, there are m machines and s resources which are specialized for one job. The sequence, in which the j Jobs will be performed as well as the assignment of machines and resources must be defined. If the sequence of the tasks is fixed, it is a flow shop Problem, if it must be defined as part of the production scheduling, it is an open shop problem.

The sequence, in which the jobs are performed, is a permutation, therefore there are $j!$ different possibilities. Furthermore, there are m possibilities to choose a machine and s possibilities to choose a resource. Therefore, the number of possibilities for a flow shop scheduling problem $k_{Flow\ Shop}$ is $k_{Flow\ Shop} = m * s * j!$. In an open shop problem, the sequence of the tasks has also be defined during the production scheduling. Therefore, the complexity is enlarged by an additional permutation: $k_{Open\ Shop} = m * s * j! * t!$.

To show the influence of the different Parameters, in Fig. 1 the number of possible solutions for an open shop problem is shown, if only the number of machines or jobs is varied. The numbers, which are not varied, are one.

It is obvious, that the influence of the number of jobs is much higher than the number of machines. The reason for this is that the number of jobs, as well as the number of tasks, enter the complexity with their faculty. A main driver of the complexity of production scheduling problems is the number of jobs to be scheduled, and therefore the time planned in the future. For open shop problems, the number of tasks, which must be performed to finish a Job, also have a great influence since it is also going into the complexity with the faculty.

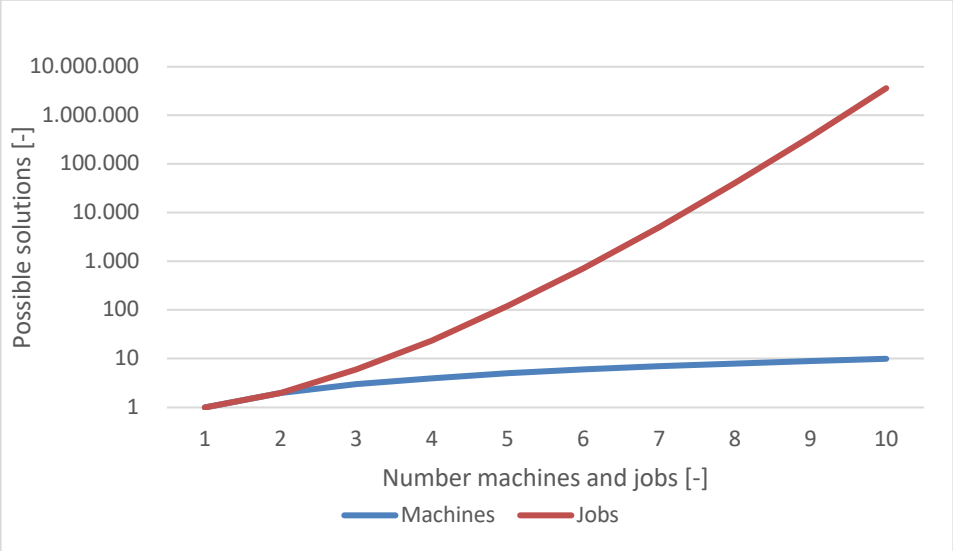

Fig. 1: Varying the number of machines and jobs for an open shop problem.

3 Reinforcement Learning

In Reinforcement Learning (RL), an agent interacts with an environment. Based on the information about the current state of the environment S, it decides which action A it will

perform. The environment changes according to the action and the agent receives a Reward R. This reward is used to train the agent, so it can make better decisions [Di20]. To differentiate the RL-algorithms, a couple of classes will be defined. Zhang and Yu differentiate between model based and model free algorithms. For the model based algorithms, there are some algorithms where the model is given, such as Alpha Go [Si16], where the rules of Go are specified. Other model based algorithms, for example Imagination-Augmented Agents (I2As) [Ra17] learn a model of the environment by themselves [ZY20].

The algorithms, without a model of the environment, are divided in the value-based and the policy based algorithms [ZY20]. Value based algorithms, like DQN [Mn15], try to optimize the action-value function, so the optimal policy is to always choose the action with the highest action value. On the opposite, the policy based algorithms, like Proximal Policy Optimization (PPO) [Sc17], directly optimize the policy, which can be seen as a set of rules that define the action to be performed in a given situation.

Furthermore, there are so-called actor-critic algorithms, which combine both approaches by using a value-based algorithm to learn a value function and a policy-based algorithm to learn the policy function [ZY20, Di20]. One example is the Asynchronous Advantage Actor-Critic (A3C) algorithm [Mn16]. Since it is a combination of value- and policy-based approaches, it will form a new category for this paper.

Zhang and Yu assign evolutionary algorithms to the model free policy based algorithms [ZY20]. Otherwise, Sutton and Barto argue that there are fundamental differences between evolutionary algorithms and other Reinforcement Learning algorithms. For example, they do not notice, which states an individual passes through during its lifetime, or which states it selects [SB18]. Nevertheless, they also have a lot in common with other RL- algorithms, which is why they are included in this paper but as an own category and not as part of the policy-based algorithms.

The selected papers will be categorized in the five categories stated in Fig. 2. Also, different benefits and shortcomings of the algorithm categories are stated, derived from literature.

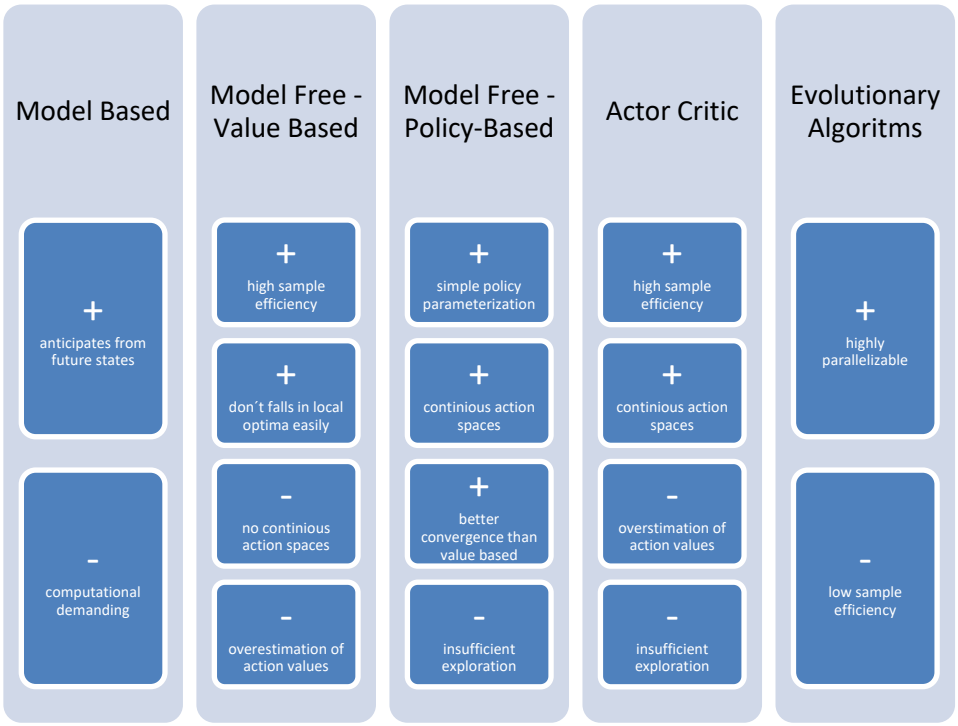

Fig. 2: Categories of Reinforcement Learning algorithms used in this paper, with their benefits and shortcomings.

Every category of Algorithm has different benefits and shortcomings. For example, an model based approach needs a planning algorithm, which is often computational demanding. Therefore, the time constraints have to be taken into accounts for real-time-decision-making [Fr18]. But if it is suitable, it has the advantage of anticipating from future states and rewards in advance [ZY20]. Model free value based methods usually cannot handle continuous action spaces and tend to overestimate the value of an action, but have a high sample efficiency and don't fall in local optima easily [ZY20]. Value based methods have the advantage of simpler policy parameterization, better convergence and are suitable for continuous action spaces [ZY20]. Actor-critic methods, which combine value- and policy-based algorithms, do also combine some of their benefits and shortcomings. For example, they have a high sample efficiency and are suitable for continuous action spaces. But they also have the problem of the overestimation of the value of an action and suffer from insufficient exploration [ZY20]. Evolutionary algorithms are on the one hand less sample efficient than other RL-algorithms, but highly parallelizable on the other hand [Sa17].

4 Technology Readiness Level for Machine Learning

Based on the recognized Technology Readiness Level (TRL), Lavin et al. developed the Machine Learning Technology Readiness Level (MLTRL) which concretizes the requirements of the TRL specifically for Machine Learning technologies [La22]. The aim of this paper is to evaluate, which MLTRL can be assigned to the different Reinforcement Learning algorithms for simulation optimization for production scheduling. Tab. 1 gives a short overview over the different MLTRLs up to MLTRL 4. Higher MLTRLs than 4 indicate, that the integration in a commercial application has started, which is not the case for simulation optimization in production scheduling.

MLTRL	Methodology	Data	Review
0	Literature research, mathematical principles, whiteboarding of concepts and algorithms.	Review of data availability	Team- or laboratory lead
1	Analysis of the model- and algorithm properties	At least representative synthesized data	Research team
2	Application in a testbed	Benchmark-data, partially or completely simulated data	Documented and reproduceable achieving of the research claims
3	Ensure interoperability, maintainability, extensibility and scalability	At least according to MLTRL2	Experts for applied AI and engineering should be involved
4	Demonstration in a real-world scenario	Representative, real data	Consideration of security and privacy aspects

Tab. 1: Overview over the MLTRL according to [La22]

For every MLTRL, there is a methodology, a data basis and a review specified. The first MLTRL is Level 0. This stage might be for example greenfield AI research. This is usually done by literature research or whiteboarding of concepts and algorithms. Often, there are no Data available, which is why at this stage, the data availability should be reviewed. Furthermore, at this early stage, the research team- or laboratory lead decides if the MLTRL should be assigned.

To bring the technology to the next level, low-level experiments to analyze the specific model or algorithm properties should be done, rather than end-to end runs for performance benchmark score. At least representative synthesized data should be used, and the research team reviews the first experiments, deciding if further experiments should be done.

To reach MLTRL 2, the Proof of Principle should be done, which means running the algorithm in a test bed, for example in a simulated environment with simulated data. Nevertheless, if available, benchmark data can be used. The research claims made in previous stages needs to be satisfied with analysis well documented and reproducible.

MLTRL 3 is the system development: The Algorithm now does not stand on its own but is prepared to be integrated in a real-world scenario as part of MLTRL4. Therefore, the code must be developed towards interoperability, reliability, maintainability, extensibility and scalability to reach prototype character, data flow and interfaces must be considered. The data are in general consistent with MLTRL 2, but at the review, teammates from applied AI and engineering should be included.

The last MLTRL considered in this literature research is MLTRL 4, the Proof of Concept. This means, that the Technology is demonstrated in a real scenario with real and representative data. During the review, besides evaluating the data quality, validity and availability, security and privacy considerations should be done.

A more detailed description of the different Levels can be found in the original Paper from Lavin et al. [La22].

5 Literature Research

This chapter will describe the performed literature research. To focus only on Reinforcement Learning algorithms for simulation optimization, the keywords “Reinforcement Learning” and “simulation optimization” are included into the search. Furthermore, to focus on the area of production scheduling, one of the following keywords, which are introduced in the definition of production scheduling in Chapter 1, must be included: “job shop”, “open shop”, “flow shop”, “shop floor” or “production scheduling”. Therefore, the following search term was entered at Google Scholar to perform the literature research:

"Reinforcement Learning" AND "simulation optimization" AND (“job shop” OR “open shop” OR “flow shop”, “shop floor” OR “Production Scheduling”)

This leads to 202 results while only considering papers which were published 2022 or later. First, papers which were not accessible, duplicates and papers with a title that does not fit the topic being discussed, were sorted out. After this, 48 Papers were left. Of these Papers, 31 were sorted out after a more detailed review, because they did not use an RL-Algorithm, were not published in a peer viewed paper or conference paper or did not handle with a problem in a production setting. After this, 17 results were left.

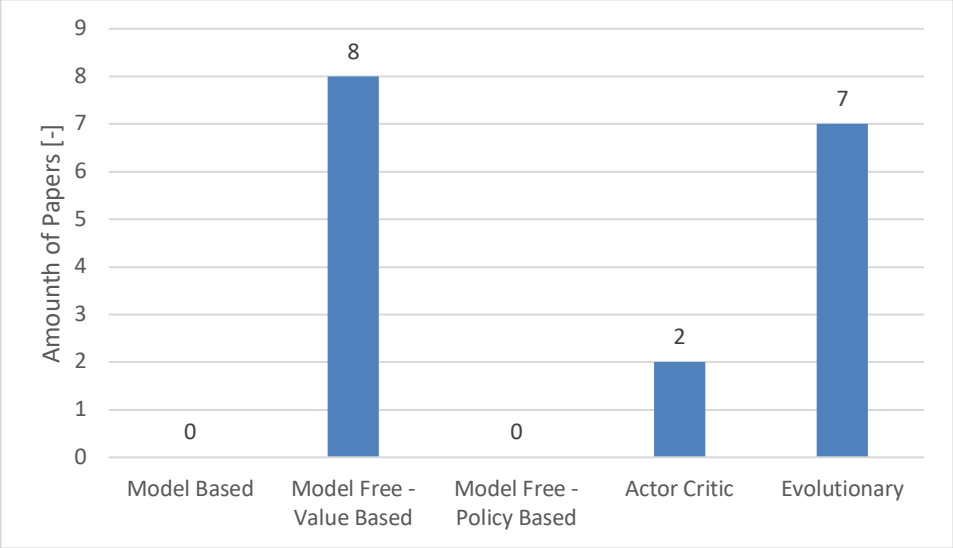

Fig. 3: Amount of papers per category.

As indicated in Fig. 3, no papers in which a model based or a model free policy-based algorithm was used could be found, but there are eight papers that deal with model free value-based algorithms, two with actor critic and seven with evolutionary algorithms. In the following Subchapter, the found papers are briefly summarized. After that, the literature is further analyzed to assign a MLTRL.

5.1 Summary

Devanga et al., Du et al., Schneckenreither et al., Wang et al., Inal et al., Wei et al. and Joo et al. have proven the principal of a model free value-based algorithms in a simulated test bed while using simulated data [DBD22, Du22, Sc22, Wa22, İn23, We22, JJS22]. Furthermore, Kuhl et al. and Zhang et al. have developed a framework, how to implement a model free value-based algorithm in a real scenario. Kuhl et al. for a warehouse system [Ku22] and Zhang et al. to assign autonomous guided vehicles for material handling in a production logistic [Zh22]. Since they have not performed an experimental study, there are no data used.

Julati et al., as well as Song et al. have proven the principle of an actor critic algorithm in a simulated test bed while using simulated data. Julati et al. have scheduled in a high mix, low volume manufacturing facility [Ju22] and Song et al. in a biopharmaceutical production process [So23].

Regarding evolutionary algorithms, Aibi and Olfa, Bergmann, Cao et al., Ma et al. and Ghasemi et al. have proven the principle of an algorithm in a simulated test bed [ADE23,

Be22a, Ca22, MZS22, GKH22]. Wurster et al. have compared different algorithms in a simulated test bed [Be22b]. They used benchmark-, simulated and real-world data. Furthermore, Panigrahi et al. have developed a framework to integrate an evolutionary algorithm in a real scenario. The aim is the production scheduling in a semiconductor wafer fabrication [Pa22].

After this short summarization, in the next chapter an MLTRL will be assigned to the different algorithms, based on the information extracted from the literature.

5.2 Analysis of the Literature

To assign the MLTRL to the different algorithms, the methodology of the reviewed papers is illustrated in Fig. 4, ordered by algorithm.

Most of the papers deal with model free value-based algorithms or evolutionary algorithms (see also Fig. 2). While the majority of these papers prove the principle of the algorithm in a simulated test bed, for two model free value based and one evolutionary algorithm a framework for a real scenario was developed. For actor critic algorithms, the principle was only proven in a simulated test bed.

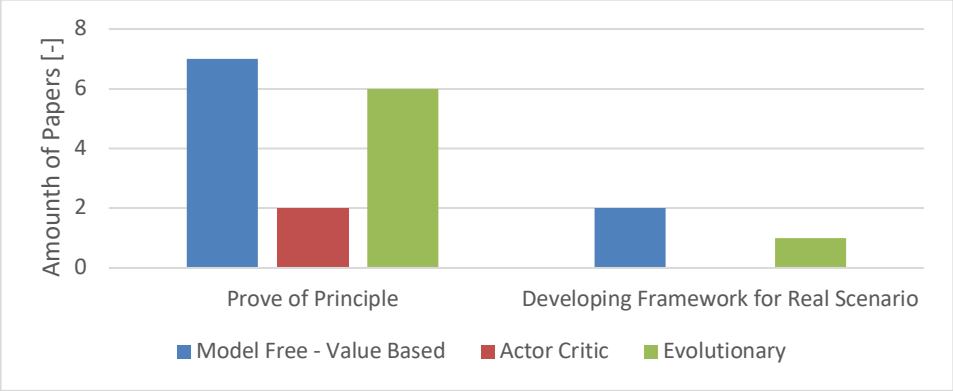

Fig. 4: Methodology of the reviewed Papers.

The Proof of Principle in a simulated test bed corresponds with a MLTRL 2. Furthermore, the peer viewed publication can be seen as a documented and reproducible documentation of the research claims and the use of simulated data also meets the requirements for MLTRL2. Therefore, all three algorithms can be seen as at least MLTRL 2.

The development of a framework for a real scenario is the requirement for MLTRL 3. Since there are no further requirements on the Data and developing the Framework requires the inclusion of applied AI and engineering, the authors of this paper argue to assign the model free value-based algorithms and the evolutionary algorithms an MLTRL 3.

As shown in Chapter 2, production scheduling is a very complex problem, especially if a lot of jobs have to be scheduled. To evaluate the possible performance, the papers in which a problem with at least 10^{10} possible solutions is handled are shown in more detail.

There are four papers with a problem with a complexity of more than 10^{10} possible solutions. The one with the smallest problem was written by Devanga et al. The performance of two value-based RL-methods, namely DDQN and DQN was compared to two dispatching rules, namely First-In-First-Out (FIFO) and Shortest Setup-and-Processing-Time (SSPT). 30 jobs have to be scheduled on one machine. Therefore, there are $30! = 2{,}65 * 10^{32}$ possible solutions. The agent assigns priority values to the incoming jobs which determine the sequence, in which the jobs are processed. The DDQN outperformed the DQN agent and the FIFO-dispatching rule. With 10 possible priority values, its performance was similar to SSTP. Nevertheless, the Devanga et al. argue, that it has the potential to outperform SSTP [DBD22].

Julati et al. have scheduled N independent jobs to M machines. The deep deterministic policy gradient with separate handling (SSDDPG) agent assigns a assignment priority to the tasks, and a free machine will process the job with the highest priority. This priority is composed of weights of different heuristics. The target is to minimize the delay past the due date, weighted by a job priority. The performance is compared to different dispatching rules, namely earliest weighted due date (EWDD), maximum mean on-time (MMOT) and shortest processing time (SPT). For the smallest problem, 35 Jobs are assigned to 10 Machines, which leads to a total of $10 * 35! = 1{,}03 * 10^{41}$ possible solutions, SSDDPG has a 14 % lower weighted delay than the best performing scheduling rule, which is MMOT. The biggest problem is to schedule 100 Jobs to 35 Machines, leading to a total of $35 * 100! = 3{,}3 * 10^{159}$ possible solutions. In this case, SSDDPG had also the best performance with a 5 % lower weighted delay than the best performing scheduling rule, which is also MMOT [Ju22].

Du et al. have used a DQN agent to schedule the crane, which transport the jobs between the machines. The agent chooses a dispatching rule, every time a transport procedure is necessary. In the most complex problem, 100 jobs and 10 machines have to be handled, leading to a complexity of $10 * 100! = 9{,}3 * 10^{158}$ possible solutions. The DQN agent outperforms a lot of different dispatching rules by far, for example FIFO, Machine Remaining Process Time (MRT) and Shortest Setup Time (SSU) to name just a few [Du22].

Ma et al. have used two evolutionary algorithms to schedule 2500 jobs to a maximum of 50 machines. The algorithms choose dispatching rules which are used to define the sequence, a machine processes the jobs in the queue. With 2500 jobs to schedule, it is obvious that the number of possible sequences tends to infinity. The best performing evolutionary algorithm leads to an 94 % reduction of tardy jobs compared to the best performing scheduling rule [MZS22].

6 Discussion

To support practitioners in deciding which Reinforcement Learning algorithm for simulation optimization for production scheduling is suitable for an implementation under real-life conditions, this paper firstly gives an overview over the topic of Reinforcement Learning and a classification, to order different algorithms. Namely, the categories model based-, model free value or policy based-, actor critic- and evolutionary algorithms are used. Since the subsequent literature research shows significant differences in the number of papers per category, it is assumed that the categories were chosen appropriately. Secondly, a short introduction into the MLTRL is given, which is an indicator of how ready a technology is for a commercial application. Afterward, the literature research was performed, to give an overview over the Technology Readiness of the different algorithms.

One criterion to assign an MLTRL is the performed review. For MLTRL2, a documented and reproducible achieving of the research claims is needed. Since a peer viewed publication can be seen as documented and reproducible, for this level the review criterion can easily be checked by a literature research. But for MLTRL3 this is not the case, since the expertise of the member of the review team is relevant. Therefore, it is assumed, that the expertise is needed to perform the corresponding methodology and to peer view the publication.

It also has to be mentioned, that a literature research cannot review in-house research and development of companies, since this information is usually confidential. So, a higher MLTRL than assigned in this paper cannot be excluded.

One result of the literature research is that no publications of model based or model free policy based algorithms could be found. One reason could be, that the literature research only takes publications into account, which were published in 2022 or later. This is a relatively short period of time, and it is possible, that there are publications which were not considered since they are older. This should be evaluated in further research. The same is for publications, in which a framework for a real scenario of an actor critic algorithm is developed. Nevertheless, by far the most publications deal with model free value based or evolutionary algorithms, so both seem to be in focus of the development. Possible reasons might be, that research has shown that other algorithms are not applicable to this problem, or just wasn't considered yet. In any case, an MLTRL3 is assigned to both algorithms, which means the next higher Technology Readiness Level can be achieved by demonstrating the technology in a real-life scenario.

When reviewing the paper which handle with a problem with at least 10^{10} possible solutions, these algorithms have comparable results to widely applied dispatching rules or are outperforming them. It is interesting to see, that none of these try to directly schedule a batch of Jobs but to assign priorities or choose a heuristic to be used to schedule the Jobs. For example in the Case of Devanga et al. 30 Jobs have to be scheduled on one machine [DBD22], which leads to $10! = 2{,}65 * 10^{32}$ possible solutions. If every incoming Job one of ten priority values is assigned, there are 10^{30} possible combinations of priority

values, which is nearly the same complexity as scheduling the Jobs directly. A direct scheduling of the Jobs seems to be possible, and its performance should be reviewed, either by literature research or by implementing and testing the algorithm.

It can also be said, that for all algorithms, for which papers were found, implementations performed better than the dispatching rules used as benchmark. A direct comparison of these algorithms cannot be done based on the literature, since the scheduling problems and benchmark-dispatching rules varying from paper to paper.

7 Conclusion

To implement a Reinforcement Learning algorithm for simulation optimization in production scheduling, the following findings of this paper should be considered:

- Current research focuses mainly on model free value based and evolutionary algorithms. Both algorithms a MLTRL 3 is assigned, therefore the next step in technology development is the demonstration in a real scenario.
- Very few publications could be found, which cover actor critic algorithms and none that cover model based or model free policy-based algorithms. The reason for this should be evaluated, since it is possible, that these algorithms are not applicable or just have not been considered yet.
- For the algorithm categories value based, evolutionary and actor-critic there is evidence, that all of these can perform significantly better than widely used dispatching rules. But there is no direct comparison between these algorithms.

Therefore, the recommended implementation strategy is to check, why there are so few recent papers which deal with actor critic and none which deal with model based and model free policy-based algorithms. Eventually, more research should be done to evaluate the usability of these algorithms for simulation optimization for production scheduling. If this has already been done, model free value based and evolutionary algorithms seem to be ready to be proven in a real scenario, so further research should be done to reach this goal. Also, a direct comparison between different RL-algorithms to evaluate their performance for different scheduling problems could be helpful to choose the best suited algorithm for a specific used case.

Acknowledgement

This work is supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant No. 01MD22001C as part of the “edge data economy initiative”.

Literaturverzeichnis

[ADE23] Aribi, D.; Driss, O. B.; El Haouzi, H. B.: Multi-Objective Optimization of the Dynamic and Flexible Job Shop Scheduling Problem Under Workers Fatigue Constraints. In (Rocha, A. P.; Steels, L.; van Herik, H. J. den Hrsg.): ICAART 2023. Proceedings of the 15th International Conference on Agents and Artificial Intelligence February 22-24, 2023, Lisbon. SciTePress - Science and Technology Publications, Setúbal, S. 301–308, 2023.

[Be22a] Bergmann, S.: Optimization of the Design of Modular Production Systems. In (Feng, B. et al. Hrsg.): 2022 Winter Simulation Conference (WSC). 11-14 Dec. 2022. IEEE, Piscataway, NJ, S. 1783–1793, 2022.

[Be22b] Behrendt, S. et al.: Extended Production Planning of Reconfigurable Manufacturing Systems by Means of Simulation-based Optimization. In (Herberger, D.; Hübner, M. Hrsg.): Proceedings of the Conference on Production Systems and Logistics: CPSL 2022. Hannover publish-Ing, S. 210–220, 2022.

[Bl19] Blazewicz, J. et al.: Handbook on Scheduling. From Theory to Practice. Springer Naure Switzerland, Cham, 2019.

[Ca22] Cao, Z. et al.: Two-stage genetic algorithm for scheduling stochastic unrelated parallel machines in a just-in-time manufacturing context. IEEE Transactions on Automation Science and Engineering 2/20, S. 936–949, 2022.

[DBD22] Devanga, A.; Badilla, E. D.; Dehghanimohammadabadi, M.: Applied Reinforcement Learning for Decision Making in Industrial Simulation Environments. In (Feng, B. et al. Hrsg.): 2022 Winter Simulation Conference (WSC). 11-14 Dec. 2022. IEEE, Piscataway, NJ, S. 2819–2829, 2022.

[Di20] Ding, Z. et al.: Chapter 2. Introduction to Reinforcement Learning. In (Dong, H.; Ding, Z.; Zhang, S. Hrsg.): Deep Reinforcement Learning. Fundamentals, Research and Applications. Springer Singapore; Imprint Springer, Singapore, 47 - 122, 2020.

[Du22] Du, Y. et al.: A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times. Transactions on Neural Networks and Learning Systems, 1-15, 2022.

[Fr18] François-Lavet, V. et al.: An Introduction to Deep Reinforcement Learning. Foundations and Trends® in Machine Learning 3-4/11, S. 219–354, 2018.

[GKH22] Ghasemi, A.; Kabak, K. E.; Heavey, C.: Demonstration of the Feasibility of Real Time Application of Machine Learning to Production Scheduling. In

(Feng, B. et al. Hrsg.): 2022 Winter Simulation Conference (WSC). 11-14 Dec. 2022. IEEE, Piscataway, NJ, S. 3406–3417, 2022.

[Gy22] Gya, R. et al.: Digital Twins. Adding intelligence to the real world. https://www.capgemini.com/gb-en/wp-content/uploads/sites/3/2022/05/Capgemini-Research-Institute_DigitalTwins_Web.pdf, Stand: 24.4.2022.

[İn23] İnal, A. F. et al.: A Multi-Agent Reinforcement Learning Approach to the Dynamic Job Shop Scheduling Problem. Sustainability 10/15, S. 1–24, 2023.

[JJS22] Joo, T.; Jun, H.; Shin, D.: Task Allocation in Human–Machine Manufacturing Systems Using Deep Reinforcement Learning. Sustainability 4/14, S. 1–18, 2022.

[Ju22] Julaiti, J. et al.: Stochastic parallel machine scheduling using reinforcement learning. Journal of Advanced Manufacturing and Processing 4/4, 1-17, 2022.

[KD21] Kumar, A.; Dimitrakopoulos, R.: Production scheduling in industrial mining complexes with incoming new information using tree search and deep reinforcement learning. Applied Soft Computing 110, S. 1–15, 2021.

[Ku22] Kuhl, M. E. et al.: Warehouse Digital Twin: Simulation Modeling and Analysis Techniques. In (Feng, B. et al. Hrsg.): 2022 Winter Simulation Conference (WSC). 11-14 Dec. 2022. IEEE, Piscataway, NJ, S. 2947–2956, 2022.

[La17] Lamghari, A.: Mine Planning and Oil Field Development: A Survey and Research Potentials. Mathematical Geosciences 3/49, S. 395–437, 2017.

[La22] Lavin, A. et al.: Technology readiness levels for machine learning systems. Nature communications 1/13, S. 1–19, 2022.

[LTD22] Luo, D.; Thevenin, S.; Dolgui, A.: A state-of-the-art on production planning in Industry 4.0. International Journal of Production Research 19/61, S. 6602–6632, 2022.

[Mn15] Mnih, V. et al.: Human-level control through deep reinforcement learning. Nature 7540/518, S. 529–533, 2015.

[Mn16] Mnih, V. et al.: Asynchronous Methods for Deep Reinforcement Learning. In (Balcan, M. F.; Weinberger, K. Q. Hrsg.): Proceedings of The 33rd International Conference on Machine Learning. PMLR, New York, New York, USA, S. 1928–1937, 2016.

[Mo20] Mourtzis, D.: Simulation in the design and operation of manufacturing systems: state of the art and new trends. International Journal of Production Research 7/58, S. 1927–1949, 2020.

[MZS22] Ma, H.; Zhang, C.; Shi, Z.: A Simulation Optimization-Aided Learning Method for Design Automation of Scheduling Rules: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). 20-24 Aug. 2022. IEEE, Piscataway, NJ, S. 1992–1997, 2022.

[Pa22] Panigrahi, S. et al.: Production Scheduling of Semiconductor Wafer Fabrication Facilities Using Real-Time Combinatorial Dispatching Rule. In (Cioboată, D. D. Hrsg.): International Conference on Reliable Systems Engineering (ICoRSE)-2021 // International Conference on Reliable Systems Engineering (ICoRSE) - 2021. Springer International Publishing AG, Cham, S. 78–90, 2022 // 2021.

[PBG22] Panzer, M.; Bender, B.; Gronau, N.: Neural agent-based production planning and control: An architectural review. Journal of Manufacturing Systems 65, S. 743–766, 2022.

[Ra17] Racanière, S. et al.: Imagination-Augmented Agents for Deep Reinforcement Learning. In (Luxburg, U. von et al. Hrsg.): Advances in neural information processing systems 30. 31st Annual Conference on Neural Information Processing Systems (NIPS 2017) Long Beach, California, USA, 4-9 December 2017. Curran Associates Inc, Red Hook, NY, S. 5694–5705, 2017.

[Ri12] Righi, R. d. R.: Preface. In (Righi, R. d. R. Hrsg.): Production Scheduling. InTech, S. X, 2012.

[Sa17] Salimans, T. et al.: Evolution Strategies as a Scalable Alternative to Reinforcement Learning. ArXiv 1703.03864v2, 2017.

[SB18] Sutton, R. S.; Barto, A.: Reinforcement learning. An introduction. The MIT Press, Cambridge, Massachusetts, London, England, 2018.

[Sc07] Schulz, A. et al.: Simulation in der operativen Produktionsplanung – Erfolgsfaktoren für KMU. Zeitschrift für wirtschaftlichen Fabrikbetrieb 1-2/102, S. 32–36, 2007.

[Sc17] Schulman, J. et al.: Proximal Policy Optimization Algorithms. ArXiv 1707.06347, 2017.

[Sc22] Schneckenreither, M. et al.: Average reward adjusted deep reinforcement learning for order release planning in manufacturing. Knowledge-Based Systems 247, S. 1–16, 2022.

[Si16] Silver, D. et al.: Mastering the game of Go with deep neural networks and tree search. Nature 7587/529, S. 484–489, 2016.

[So23] Song, W. et al.: Stochastic Economic Lot Scheduling via Self-Attention Based Deep Reinforcement Learning. IEEE Transactions on Automation Science and Engineering, S. 1–12, 2023.

[Wa22] Wang, X. et al.: Digital Twin-Assisted Efficient Reinforcement Learning for Edge Task Scheduling: 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring). IEEE, S. 1–5, 2022.

[We22] Wei, Q. et al.: A Self-Attention-Based Deep Reinforcement Learning Approach for AGV Dispatching Systems. IEEE Transactions on Neural Networks and Learning Systems, S. 1–12, 2022.

[Zh22] Zhang, L. et al.: Reinforcement learning and digital twin-based real-time scheduling method in intelligent manufacturing systems. IFAC-PapersOnLine 10/55, S. 359–364, 2022.

[ZY20] Zhang, H.; Yu, T.: Chapter 3. Taxonomy of Reinforcement Learning Algorithms. In (Dong, H.; Ding, Z.; Zhang, S. Hrsg.): Deep Reinforcement Learning. Fundamentals, Research and Applications. Springer Singapore; Imprint Springer, Singapore, S. 125–133, 2020.