Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

DYNAMICALLY ADJUSTING SEQUENCING RULES IN A COMPLEX MANUFACTURING SYSTEM WITH UNCERTAINTY

Thomas Voss

Leuphana University
Universitaetsallee 1
Lueneburg, 21335, GERMANY

ABSTRACT

Especially in complex manufacturing systems and uncertain conditions, sequencing operations in a machines' queue can pose a difficult problem. Decentral approaches have proven useful, often the usage of sequencing rules has been a viable option. Still, no rule can outperform all other rules under varying system performance. For that reason, reinforcement learning (RL) is used as a hyper-heuristic to dynamically select and adjust sequencing rules based on system status. The trained agent is tested in a complex manufacturing system with uncertainty and evaluated under various conditions matching and outperforming the best sequencing rules found in a preliminary simulation study. Given an unknown scenario, it has to be evaluated if the RL hyper-heuristic is also able to change the sequencing rule suitable for the scenario, providing a robust performance.

1 INTRODUCTION

Given that the problem of sequencing operations in a complex manufacturing system is usually NP-hard, heuristics need to be applied to find results in a feasible amount of time. For that reason, intelligent, adaptable, and autonomous systems have been developed to sequence operations under uncertainty within large manufacturing systems. The use of reinforcement learning (RL) has proven to be suitable for selecting specific operations from queues or repairing schedules. Even though the method works for small scenarios, it is very difficult to assess the impact of picking a particular order for processing in larger systems. Therefore, the dissertation makes a novel contribution to the literature by presenting RL as hyper-heuristic dynamically adjusting simple priority based sequencing rules. Once trained, the agent behavior is ported to other scenarios to evaluate if robust performance under unknown circumstances can be achieved.

2 DYNAMICALLY ADJUSTING SEQUENCING RULES

Simple sequencing rules assign a priority to the jobs waiting in the queue depending on given criteria whenever a machine is freed for operation. One example could be sorting the jobs by the "shortest processing time" (SPT) or the "earliest due date" (EDD) to choose the one with the highest priority to be processed next. This method of sequencing jobs in front of a machine is long known and still applied in the industry due to its simplicity. Commonly known sequencing rules have been reviewed by Panwalkar and Iskander (1977), who did an extensive study presenting more than 100 rules. These rules can be applied to a wide range of cases and are not scenario specific. Given the fact, that no rule outperforms all the others under varying system loads, the idea of dynamically switching and adjusting the rules to the current situation has been pursued. For that reason, data from simulation runs is documented and machine learning techniques, such as neural networks (NN) are applied afterward to estimate the performance. Mouelhi-Chibani and Pierreval (2010) use a NN to dynamically switch rules on every machine depending on the current system state. Their scenario consists of only two machines and the set of dispatching rules consists of the SPT and EDD rule. Heger (2014) used NN as well as Gaussian process regression to forecast system performance and dynamically select and adjust sequencing rules. Still, the results of the regression were documented in a table prior to the online testing and not generated dynamically. Additionally, the performance data is usually recorded at the end of a simulation run and thereby neglecting the impact of uncertainty over the simulation run. Based on the simulation of the complex scenarios an approach utilizing simulation, computation power, and hyper-heuristics as well as data processing technique is applied. RL uses supervised and unsupervised learning and therefore is another type of machine learning. Starting from an unknown environment, an agent can perform different actions from a selection of different activities during the simulation (also called policy), based on the current environment variables (also called states). Through the interaction with the environment and the resulting change, the agent receives a reward for the activity carried out. By trying different actions, the agent learns over time which activities bring a reward. Evaluated state-action-pairs emerge, which are stored in a Q-Table or NN respectively.

3 APPLICATION IN A MANUFACTURING SYSTEM

Using the discrete event-based simulation, realized with AnyLogic^{TM}, an RL agent is trained as hyper-heuristic in a flexible job shop scenario. The scenario utilizes multiple machines with restrictions regarding precedence, preemption, and set up. Different product families are being manufactured, resulting in setup tasks between them. To increase complexity all jobs undergo processing on different machine groups multiple times, based on randomly generated sequences, to represent re-entrant processes. The processing times are drawn from a uniform distribution whereas the setup times are sequence-dependent and static. This results in a highly flexible and complex manufacturing system with uncertainty. The key performance indicators (KPIs) are usually documented after each simulation run. Given the fact, that RL can learn not only based on the outcome but train during the simulation, it is being evaluated, if the approaches provided by the literature can be outperformed in a complex scenario. Additionally, it is being evaluated, if the increased amount of information generated during the training, can enhance the precision and accuracy of the dynamic adjustment and increase the performance of the system, given the stochastic variation in the system. During the training different reward functions, amount of observations as well as actions are being tested to find the most suitable parameters. Given the trained policy on one system mentioned above, it is being evaluated if the transfer of knowledge from one system to another is possible with RL.

4 CONCLUSION

An RL hyper-heuristic has been integrated into a complex flexible job shop environment with uncertainty to dynamically select and adjust sequencing rules. During the evaluation, it has been proven, that the dynamic selection of single attribute sequencing rules can improve the key performance indicators of the system. Current results indicate, that a dynamic adjustment using more complex rules might be a viable option. The usage of RL in a more dynamic scenario will be evaluated, considering changing product mixes and varying system utilization. The analysis regarding the possibilities of knowledge transfer is still to be conducted. Further evaluation of the approaches will include the aspects of routing products.

ACKNOWLEDGMENTS

The author would like to express his gratitude to the team from Pathmind for the insightful discussion. This contribution has not received any funding from any organization.

REFERENCES

Heger, J. 2014. *Dynamische Regelselektion in der Reihenfolgeplanung*. Wiesbaden: Springer Fachmedien Wiesbaden.

Mouelhi-Chibani, W., and H. Pierreval. 2010. “Training a neural network to select dispatching rules in real time”. *Computers & Industrial Engineering* 58(2):249–256.

Panwalkar, S. S., and W. Iskander. 1977. “A Survey of Scheduling Rules”. *Operations Research* 25(1):45–61.