*Proceedings of the 2022 Winter Simulation Conference*
*B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.*

# DISPATCHING IN REAL FRONTEND FABS WITH INDUSTRIAL GRADE DISCRETE-EVENT SIMULATIONS BY USE OF DEEP REINFORCEMENT LEARNING

Patrick Stöckermann

Infineon Technologies AG
Am Campeon 1-15
Neubiberg, 85579, GERMANY

## ABSTRACT

Optimization of lot dispatching in semiconductor manufacturing is essential for time and cost efficient production. In contrast to constraint, mixed-integer, or answer set programming approaches to the problem, Reinforcement Learning (RL) is not limited to local optimization and ideally generalizes in similar domains. Although there are numerous publications utilizing RL, most are applied to small idealized test instances which are not realistic. We propose a Deep Reinforcement Learning (DRL) approach that is applied to real world scenarios. We utilize an industry grade discrete-event simulation as a training and testing environment in order to find global optima for the average tardiness and the Flow Factor (FF). Due to the current limitations of the computing hardware, no relevant results could be obtained yet. However, we expect to improve these measures compared to the highly realistic dispatching rules modelled in the simulation environment by controlling generally acknowledged high impact work centers.

## 1 INTRODUCTION

Lot dispatching is a central problem in complex job-shops such as in semiconductor manufacturing because optimizing dispatching rules is essential for an efficient Work In Progress (WIP) flow. Traditionally, heuristics are used as static dispatching rules. Another common approach is to locally optimize with combinatorial optimization techniques like as constraint, mixed-integer, or answer set programming. In practice, this can be done only locally as the entire system is highly stochastic, difficult to plan, and the problem is NP-hard. A further difficulty is the reentrant flow. This means that lots can revisit steps because there are frequent reoccurrences of processing sequences. We aim to mitigate these problems by utilizing a DRL algorithm. The Deep Neural Network of the DRL algorithm enables the learning of highly complex stochastic strategies.

## 2 RELATED WORK

Previous publications regarding the application of RL for dispatching in semiconductor manufacturing include the multi-agent approach by Waschneck et al. (2018) modelling four work centers.
Park et al. (2021) apply a Graph Neural Network (GNN) scheduler to Job-Shop Scheduling Problems (JSSPs). This is reasoned with the very good generalization capability of GNNs. Liu, Chang, and Tseng (2020) propose an RL approach utilizing an actor-critic network to solve JSSPs. Both approaches achieve good results on commonly referenced benchmark problems. However, these datasets contain no more than 20 machines.

## 3 APPROACH

The general approach of RL is the independent learning of an agent. The agent interacts with an environment to optimize its behavior with respect to received rewards through trial and error. For the application of lot dispatching, an idealized environment is usually used in the literature. In this work, however, a highly realistic environment with over a thousand equipments is used. An RL algorithm is applied to an industrial size discrete-event simulation environment based on data that is mined from real fronted fab production data. The goal is to optimize certain Key Performance Indicators (KPIs) like the average tardiness and the FF. The performance is measured by comparing it to the default dispatching rules in the simulation. The DRL approach enables the direct optimization for those KPIs through the reward function.

As the scope of the simulation is extensive, finding bottlenecks and critical elements in the WIP flow through data analysis is crucial. The optimization can be applied more effectively to these bottlenecks, such as lithography clusters. Due to the very diverse real product mix, if the agent selects lots directly, the action and observation space becomes too complicated and the convergence of the algorithm gets much more difficult. Instead of making such specific decisions, we propose that the RL agent is used exclusively for strategic WIP-balancing decisions, which also minimizes the problem that such a simulation cannot represent all details of the reality. Since this top-level approach is only applicable for large complex systems, it is difficult to test it for a very small instance. After the queuing lots are filtered according to certain constraints, the agent decides on a product group based on the state of the fab and the remaining lots. Subsequently, traditional heuristics are used to decide which lot within the product group is dispatched.

## 4 IMPLEMENTATION

To utilize the existing discrete-event simulation, an interface is used, which handles the communication between the state of the art Proximal Policy Optimization RL algorithm in Python and D-SIMLAB's D-SIMCON simulator. Every time a dispatching decision has to be made at the initially defined bottleneck tools, the simulator triggers a callback to the RL agent to decide on a product group. Most of the execution time is spent running and communicating with the simulation. It takes multiple hours to simulate one episode of two months in the fab including the RL control of 50 bottleneck machines. We expect to have to run through a three-digit number of episodes. To mitigate this problem, multiple parallel simulations with different random seeds are sampled asynchronously. Currently, the training is too slow to achieve relevant results within reasonable time as it is limited by the hardware capabilities. By scaling the parallelization of the training from a local machine to a Load Sharing Facility (LSF), we aim to solve this problem.

## 5 EXPECTED RESULTS AND OUTLOOK

We expect to show that the agent is able to learn a strategy that improves the fab level FF and average tardiness compared to the highly realistic heuristics modelled in the simulation. Furthermore, we want to prove that by optimizing the priority of product groups and combining them with heuristics at bottleneck work centers alone, the WIP can be balanced to improve fab level KPIs.

## REFERENCES

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

Park, J., J. Chun, S. H. Kim, Y. Kim, and J. Park. 2021. "Learning to schedule job-shop problems: Representation and policy learning using graph neural network and reinforcement learning". International Journal of Production Research 59(11):3360–3377.

Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek. 2018. "Deep reinforcement learning for semiconductor production scheduling". 2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC):301–306.