Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

SCALING DEEP REINFORCEMENT LEARNING FOR QUEUE-TIME MANAGEMENT IN SEMICONDUCTOR MANUFACTURING

Harel Yedidsion
Prafulla Dawadi
David Norman
Emrah Zarifoglu

AI/ML Team, Applied Materials Inc.
3050 Bowers Avenue
Santa Clara, CA 95054, USA

ABSTRACT

Queue-Time Constraints (QTCs) set a maximum waiting time for lots between consecutive process steps. In semiconductor manufacturing, exceeding these limits results in yield loss, rework, or scrapping. Managing QTCs is challenging due to the need for lots to wait until there is available capacity for the final step. Specifically, accurately calculating the capacity is computationally expensive, making it difficult to handle large instances. Our research addresses the scalability of QTC management in real fabs with numerous constraints. We propose a deep Reinforcement Learning (RL) solution to handle lot release into the QTC. We describe the infrastructure developed for RL training using actual fab data, assess the performance of our RL approach, and compare it to three baseline solutions. Our empirical evaluation demonstrates that the RL method surpasses the baselines in key performance metrics including queue-time violations, while requiring negligible online compute time.

1 SYSTEM, SIMULATOR, AND METHODOLOGY

We propose a deep RL-based approach to control a Queue-Time Management System (QMS) in a large-scale realistic problem. Previous papers dealt with smaller QMS problems, scheduling one part (Yedidsion et al. 2022) and ten parts (Yedidsion et al. 2023) using artificially generated data. This work involves managing hundreds of part types with data taken from a real fab. We designed a custom-built simulator to simulate a fab with QTCs. The simulator allows us to flexibly define a fab environment with any number of stations, station families, and lots (batches of wafers) of multiple part types. Figure 1 displays a simplified system diagram of the environment considered in this research. The QMS controls the Gate steps and decides which lots to release at each time-step. For each part type, we define a route, which is a set of processing steps. Each step is assigned to dedicated stations in a station family and has its own processing times. Any pair of steps in a route can have a QTC between them. The simulator supports releasing multiple lots of multiple part types at each time-step.

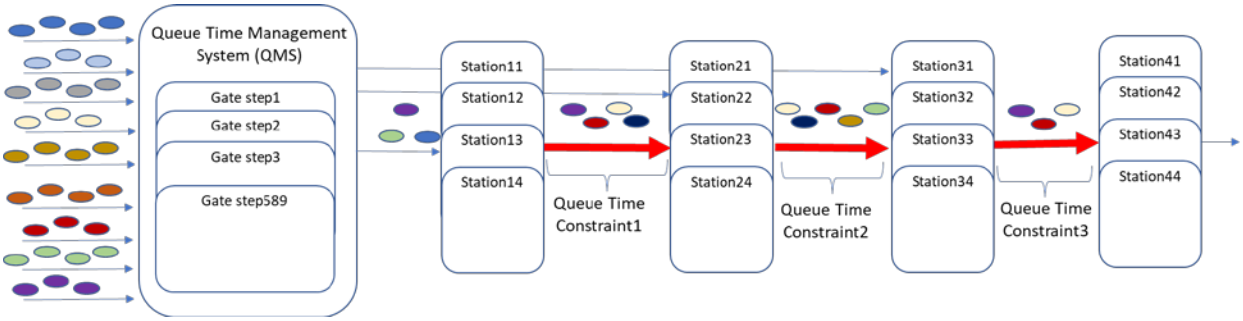

Figure 1: Simplified system diagram.

The experimental setup that was used for the evaluation is based on data from a real fab. The environment has 589 gate steps, 11 station families with 74 stations. Each part has 1-3 QTCs in its route. To create a training database, we collected 450 snapshots of Work In Progress (WIP), and recorded 450 days of lot arrivals from real fab data. When combining initial WIP, with lot arrivals, we get 202,500 unique day-long episodes for training and testing. We used a queue-time based dispatch rule that orders lots based on their remaining queue-time, so that the lot that has the least amount of time to violate its QTC will be scheduled first. Each episode has 96 time-steps, and each time-step takes 15 minutes. At each time-step the agent can take a multi-discrete action indicating how many lots to release of each part type with respect to availability. The simulator releases these lots and simulates 15 minutes of processing time. Following that, the RL agent receives an observation reflecting the system's state at the end of the time-step, and a reward. The state-action-reward sequence is saved, and periodically the RL algorithm uses this experience to update the weights of the neural network which represents the policy. The policy picks the next action and is updated to maximize the cumulative reward over the time horizon. We designed a reward structure which encourages the agent to minimize the number of queue-time violations while optimizing for throughput and the number of successful lots. For the RL algorithm, we used the Proximal Policy Optimization (PPO) algorithm (Schulman et al. 2017) implementation by https://stable-baselines3.readthedocs.io.

2 EVALUATION AND CONCLUSION

We compare the performance of the RL agent (PPO) to that of three baseline agents.

1. *Kanban* agent: Maintains a fixed queue size. Proposed in (Scholl and Domaschke 2000).
2. *Frequency* agent: This agent releases a lot at a frequency which is closest to the processing time of the second step in each route.
3. *Always* agent: The *Always* agent releases all available lots and provides an upper bound on throughput (although at the cost of many violations).

For testing, each agent controlled the gate step for 30 unique episodes, and average key performance indicators (KPIs) were recorded. The evaluation metrics for each agent are summarized in Table 1.

Table 1 - Results

	KPI			
Agent	#completions	#successes	#violations	%v/(v+s)
Always	122.83	136.43	12.63	8.47%
Freq	111.10	124.63	7.30	5.53%
Kanban	120.60	135.50	1.63	1.19%
PPO	119.47	133.96	1.30	0.96%

PPO achieves a low number of violations (less than 1% of all constraints) with a relatively high number of successes and high throughput. The model is constantly improving with more training and parameter tunning. In terms of compute time, PPO can quickly decide on the next action given a state observation, and that time does not grow exponentially with the size of the problem as opposed to exact solution methods.

REFERENCES

Scholl, W. J., and Domaschke. 2000. "Implementation of Modeling and Simulation in Semiconductor Wafer Fabrication with Time Constraints between Wet Etch and Furnace Operations". *IEEE TSM*, 13(3): 273-277.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. "Proximal Policy Optimization Algorithms". *arXiv preprint arXiv*:1707.06347, https://arxiv.org/pdf/1707.06347.pdf, accessed 28th June 2023.

Yedidsion, H, P. Dawadi, N. Norman, and E. Zarifoglu. 2022. "Deep Reinforcement Learning for Queue-Time Management in Semiconductor Manufacturing". In *Proceedings of the 2022 Winter Simulation Conference*, 3275-3284. IEEE.

Yedidsion, H, P. Dawadi, N. Norman, and E. Zarifoglu. 2023. "Optimizing Queue Time Constraint Scheduling in Semiconductor Manufacturing Using Deep Reinforcement Learning". *IEEE TSM*, Under Review.