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

DISPATCHING IN REAL FRONT-END FABS WITH INDUSTRIAL GRADE DISCRETE-EVENT SIMULATIONS BY DEEP REINFORCEMENT LEARNING WITH EVOLUTION STRATEGIES

Patrick Stöckermann

Infineon Technologies AG
Am Campeon 1-15
81726 Neubiberg, GERMANY

ABSTRACT

Scheduling is a fundamental task in each production facility with implications on the overall efficiency of the facility. While classic job-shop scheduling problems become intractable when the number of machines and jobs increases, the problem gets even more complex in the context of semiconductor manufacturing, where flexible production control and stochastic event handling are required. In this paper, we propose a Deep Reinforcement Learning approach for lot dispatching to minimize the Flow Factor (FF) of a digital twin of a real-world, stochastic, large-scale semiconductor manufacturing facility. We present the first application of Reinforcement Learning (RL) to an industrial grade semiconductor manufacturing scenario of that size. Our approach leverages a self-attention mechanism to learn an effective dispatching policy for the manufacturing facility and is able to reduce the global FF of the fab.

1. RELATED WORK

Previous approaches (Waschneck et al. 2018; Kuhnle et al. 2021) regarding the application of RL for dispatching in the Job-Shop Scheduling Problem (JSSP) setting of semiconductor manufacturing achieve good results on commonly referenced benchmark problems that are deterministic and contain no more than 20 machines. Extending on those approaches, Tassel et al. (2023) propose a self-supervised, RL-based approach to improve the global dispatching of a fab. The approach is evaluated on the academic, open-source SMT2020 testbed (Kopp et al. 2020), which models a modern wafer fab including over 1000 tools. While the scale of the addressed problem goes well beyond the small test datasets for JSSP, the SMT2020 fab models have some shortcomings compared to real manufacturing settings. In fact, our scenario's load mix is much more diverse, and the tool dedications are more complex. Our dataset contains more than ten times the number of products and features flexible processing times on different tools for the same operation.

2. APPROACH

We utilize a neural network architecture based on self-attention, which is size-agnostic regarding the length of the queue by use of a self-attention mechanism (see Figure 1). Our optimizer is based on the sample efficient Covariance Matrix Adaptation Evolution Strategies (CMA-ES) approach (Hansen and Ostermeier 2001) that considers the covariance of parameters and aims for faster convergence while sampling. The training can be highly parallelized, which is essential as our simulation requires extensive runtime.

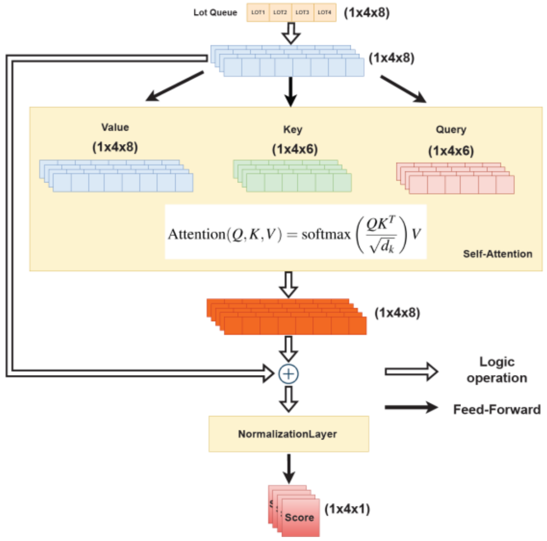

Figure 1: Neural network architecture.

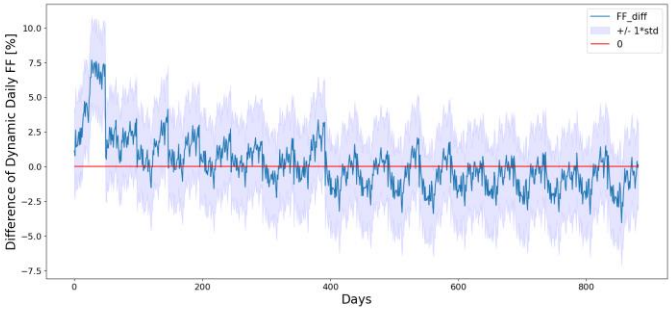

Figure 2: FF improvement during the training.

3. RESULTS

To enable good generalization, we train a model on two loading scenarios in parallel (see Figure 2). Each policy is therefore evaluated over different loading scenarios within each episode. A scenario refers to a specific interval of time from the past two years in the modeled wafer fab. In fact, different loading scenarios vary in the number of free tools, the total number of WIP lots, and the product mix. When training on two loading scenarios in parallel, we observe that, during the testing, the strategy generalizes for those two scenarios over different random seeds and previously unseen scenarios (see Table 1).

We additionally introduce a tardiness penalty as we find that the pure optimization of the FF sometimes leads to increased total tardiness. If the tardiness is higher than for the reference, the reward is the FF discounted by the ratio at which the tardiness is worse relative to the reference. Our results show that this is effective in preventing increased tardiness. Training on two sufficiently different scenarios yields a much more stable policy than training on just one with different random seeds. However, some scenarios show considerably better improvement than others. This problem can approached in the future by training on even more scenarios and random seeds in parallel.

Table 1: Evaluation of a policy trained with 64 CPU cores on loading scenarios 1 and 2 in parallel.

Relative Improvement	Scenario 1	Scenario 2	Scenario 3	Scenario 4
FF	2.72 % ± 0.34	0.88 % ± 0.28	5.00 % ± 0.22	0.60 % ± 0.31
Tardiness	36.92 % ± 1.10	22.84 % ± 1.15	26.91 % ± 1.87	17.93% ± 1.43
Completed Wafers	0.35 % ± 0.11	0.79 % ± 0.09	0.66 % ± 0.05	0.13 % ± 0.16

REFERENCES

Hansen, N., and A. Ostermeier. 2001. “Completely Derandomized Self-Adaptation in Evolution Strategies”. Evol. Comput. 9(2):159–195.

Kopp, D., M. Hassoun, A. Kalir, and L. Mönch. 2020. “SMT2020—A Semiconductor Manufacturing Testbed”. IEEE Transactions on Semiconductor Manufacturing 33(4):522–531.

Kuhnle, A., J. Kaiser, F. Theiß, N. Stricker, and G. Lanza. 2021. “Designing an Adaptive Production Control System using Reinforcement Learning”. J. Intell. Manuf. 32(3):855–876.

Tassel, P., B. Kovács, M. Gebser, K. Schekotihin, P. Stöckermann, and G. Seidel. 2023. “Semiconductor Fab Scheduling with Self-Supervised and Reinforcement Learning”. In *Proceedings of the Winter Simulation Conference*. IEEE.

Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek. 2018. “Deep Reinforcement Learning for Semiconductor Production Scheduling”. In *Proceedings of the ASMC*, 301–306: IEEE.