DOI: 10.22032/dbt.57800

Simulation in Produktion und Logistik 2023
Bergmann, Feldkamp, Souren und Straßburger (Hrsg.)
Universitätsverlag Ilmenau, Ilmenau 2023
DOI (Tagungsband): 10.22032/dbt.57476

Deep Reinforcement Learning for Workload Balance and Due Date Control in Wafer Fabs

Deep Reinforcement Learning für Workload Balance und Fälligkeitskontrolle in Wafer Fabs

Zhugen Zhou, Oliver Rose, Universität der Bunderswehr München, Munich (Germany), zhugen.zhou@unibw.de, oliver.rose@unibw.de

Abstract: Semiconductor wafer fabrication facilities (wafer fabs) often prioritize two operational objectives: work-in-process (WIP) and due date. WIP-oriented and due date-oriented dispatching rules are two commonly used methods to achieve workload balance and on-time delivery, respectively. However, it often requires sophisticated heuristics to achieve both objectives simultaneously. In this paper, we propose a novel approach using deep-Q-network reinforcement learning (DRL) for dispatching in wafer fabs. The DRL approach differs from traditional dispatching methods by using dispatch agents at work-centers to observe state changes in the wafer fabs. The agents train their deep-Q-networks by taking the states as inputs, allowing them to select the most appropriate dispatch action. Additionally, the reward function is integrated with workload and due date information on both local and global levels. Compared to the traditional WIP and due date-oriented rules, as well as heuristics-based rule in literature, the DRL approach is able to produce better global performance with regard to workload balance and on-time delivery.

1 Introduction

In semiconductor wafer fabs, dispatching rules are commonly applied as shop floor control policies to achieve specific objectives, such as workload balance and on-time delivery. Among the WIP-oriented rules, minimum inventory variability scheduling (MIVS) (Li et al. 1997) stands out as a representative method for workload balance. MIVS prioritizes operations with high WIP and downstream operations with low WIP to prevent starvation at the downstream operations. Conversely, it assigns low priority to operations with low WIP and downstream operations with high WIP. MIVS aims to keep the WIP of each operation close to the average target WIP level, which effectively reduces WIP variation and accelerates job movement.

As many wafer fabs change from mass production to mass customization to satisfy customers, due dates become another critical factor. Due date-oriented rules like operational due date (ODD) (Keskincocak and Tauyr 2004) are applied to achieve on-

time delivery. ODD rule breaks up slack time into as many segments as the number of operations of a job, which means it considers due dates for all intermediate operations. The ODD value of operation *i* is defined as: *ODD = ReleaseTime + RPT(i) * DDFF*, where *RPT(i)* denotes the raw processing time for a sequence of processing steps or operations from operation 1 to operation *i* (including operation *i*) and *DDFF* denotes target due date flow factor which is the ratio of target cycle time and raw processing time of a job.

In modern wafer fabs, achieving both WIP and due date targets simultaneously is a major challenge as workload balance and on-time delivery are conflicting objectives (Zhou 2015). WIP-oriented rules often overlook job due dates, while the due date-oriented rules prioritize tardy jobs most possibly at the expense of workload balance. To address this challenge, a heuristic-based composite rule (Zhou and Rose 2011) that combines three single rules, i.e., least work at next queue rule (LWNQ, the job with the least workload at the downstream queue is preferred), ODD rule and shortest processing time rule (SPT, the job with the shortest processing time is preferred), is developed in a previous study. A design of experiment is used to calculate the scaling parameters which determine the contribution of each single rule. The composite rule outperforms the three single rules in terms of cycle time and tardiness performance. However, its disadvantage is that the performance depends heavily on user-defined scaling parameters, making it computationally expensive to achieve globally optimal performance. Therefore, self-learning methods such as machine learning offer promising approaches for dispatch decision-making in achieving workload balance and on-time delivery.

Deep reinforcement learning (DRL) has attracted more and more attention for decision-making problems in recent years (Panzer et al. 2021). Riedmiller and Riedmiller (1999) proposed a RL approach in which agents deal with specified features of the factory to optimize global sum tardiness. Similarly, Waschneck et al. (2018) applied Google DeepMind's deep-Q-network agent algorithm at an abstract frontend-of-line semiconductor production facility taking due date as global optimization objective. Sakr et al. (2021) proposed an application of DRL for dispatching and resource allocation in a real semiconductor manufacturing system. Both of the input states and reward function consist of local and global information of the wafer fabs. Different from the literature, our study aims to find out if the DRL is able to solve the problem of conflicting objective, i.e., workload balance and on-time delivery, for wafer fabs. In this paper, an independent dispatch agent is defined to achieve self-learning for dispatch decision. Firstly, the agent is responsible for observing the state changes of wafer fabs. Then it learns to take action for dispatch decision by employing the deep-Q-network reinforcement learning. After that it receives rewards, which are the key performance indicators (KPIs) in wafer fabs, to correct its action. The agent continuously learns and improves its decision-making abilities in order to select the most optimal dispatch decision.

2 Simulation Environment and Deep Reinforcement Learning

2.1 Problem Statement

Semiconductor wafer fabs are considered as one of the most complex manufacturing systems because of its production variations such as mix products, re-entrant flow, hundreds of work-centers, machine breakdown, setup and batch processing. To achieve expected objectives, dispatching rules are commonly used as a way of shop floor control. WIP-oriented rules reduce variability by controlling the flow of jobs, which is able to speed up job movement and reduce cycle time. However, it often balances workload at the expense of the pace of job movement, for example, some jobs finish ahead of schedule, while others become tardy. On the contrary, due date-oriented rules focus on the job movement with right pace toward due date, which often results in high WIP level and cycle time. When both low WIP level and good on-time delivery performance are desired simultaneously, it can be challenging to achieve both objectives using either a WIP-oriented rule or a due date-oriented rule alone.

2.2 Deep Reinforcement Learning

Deep reinforcement learning is a type of machine learning algorithm that involves training an artificial neural network to learn how to make decision in a dynamic environment (Panzer et al. 2021). Agents interact with the environment by taking actions with action-selection policy $\pi_t(a|s)$, and receive feedback in the form of reward which is expressed as action-value function $Q_\pi(s|a)$. The Q-function is the accumulated discount future reward when policy π is utilized. The optimal Q-function $Q^*(s|a)$ is defined as the maximum reward that can be obtained by applying the optimal policy (Equation (1)), where s is the state, a is the action, t is the time-step, γ is the discount factor with a value in [0,1], π is the action-selection policy. The agent applies artificial neural network called deep-Q-network as the Q-function approximation to predict the expected reward. Through iteratively training the deep-Q-network is able to approximate the optimal Q-function Q^*.

$$Q^*(s,a) = max_\pi \mathbb{E}\left[\sum_t r_t \,.\, \gamma^t \mid s_t = s, a_t = a, \pi\right] \tag{1}$$

2.3 Simulation Environment

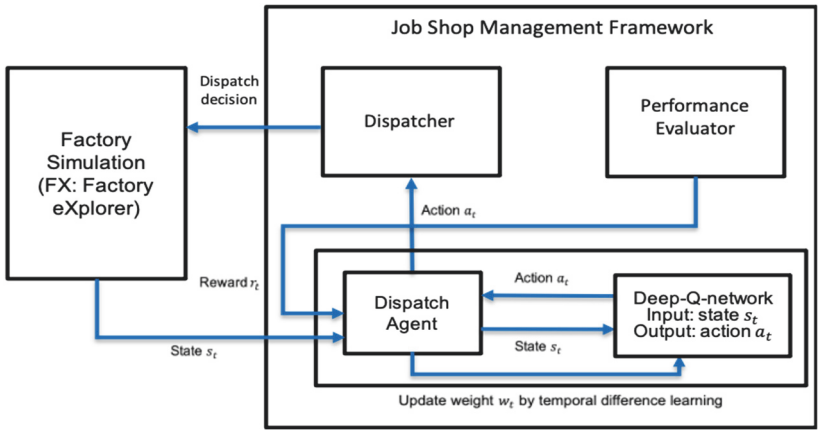

Figure 1: Simulation environment containing deep reinforcement learning

Figure 1 illustrates the simulation environment used in this study. The simulation software is Factory eXplorer (FX) which is a commercial simulation software for factories. A job shop management framework that addresses issues of workload balance and due date control (Zhou and Rose 2015) is developed. The framework comprises two components which are dispatcher and performance evaluator. The dispatcher implements various operational control strategies and calculates the job priority when a dispatch decision is required. The performance evaluator calculates KPIs to assess the performance of each strategy. To integrate the concept of DRL, a dispatch agent is introduced in this study. The agent mainly interacts with the factory simulation by observing state changes and receiving rewards. Upon detecting an available machine, the agent determines the next action (i.e., job selection) using its own deep-Q-network. Based on the factory states, the deep-Q-network predicts the optimal dispatch decision, and the agent executes it to receive rewards that are the KPIs from the performance evaluator. The agent trains the network via temporal difference learning using the differences between current and historical rewards, with the objective to predict better action over time. Table 1 presents the deep-Q-network configuration details.

Table 1: An overview of the deep-Q-network

Input state s_t (see Section 2.4)	Each work-center has two states Each job in the queue has three states
Output action a_t	Each output neuron represents the Q value of the job
Reward r_t (see Section 2.5)	Reward function consists of four parts
Topology	Input layer -> 2 Hidden layers (128 and 64 neurons) -> Output layer
Learning method	Temporal difference learning
Activation function	Rectified linear unit
Learning rate	0.01
Discounted rate	0.8

2.4 System States

The agent needs to learn to make decisions based on the changing states of the system which is the wafer fabs in this case. System states are important as they provide key information for the agent to learn accurately and take proper action. The deep-Q-network facilitates the transformation between system states and action. The system states are considered as inputs and fed into the deep-Q-network. In this study, the system states are defined based on the workload and due date, as they are the major concerns. They consist of work-center states and the states of jobs in the queue.

- Work-center states:
  - Number of its machines considering machine breakdown
  - Number of machines at its downstream work-centers (one-step-ahead) considering machine breakdown
- Job states:
  - Operation due date
  - Final due date
  - Workload indicator of its downstream work-centers (one-step-ahead) (Zhou and Rose 2019)

The work-center states are determined by the availability of machines at current work-center and its one-step-ahead downstream work-centers. The agent must learn to assess whether these work-centers have sufficient capacity to process jobs in the event of machine breakdowns. For example, if a downstream work-center is already overloaded and loses capacity due to a breakdown, the agent should not assign more jobs to it.

When considering job states, the agent takes into account both the operation due date and the final due date. The operation due date serves as a small milestone for the agent to learn whether jobs are progressing at the correct pace through the wafer fabs. The final due date represents the deadline milestone for completing a job to avoid being tardy. In addition to due dates, the agent also assesses the workload indicator, which is defined as the sum of production hours, including load/unload time and raw

processing time of the operation, at the current work-center and its one-step-ahead downstream work-centers. Based on this information, the agent can learn to choose jobs that are appropriate for the workload at downstream processes.

2.5 Reward Function

The reward function determines the reward or penalty that the agent receives based on its actions in the environment. Therefore, the reward function is designed to encourage the agent to take actions that lead to positive outcomes and avoid actions that lead to negative outcomes. The reward function is similar to the system state in that it incorporates feedback from both work-centers and jobs at both local and global levels, as shown in Equation (2).

$$Reward_{i,t} = WD_{j,one-step-ahead\_j} + QT_{i,t} + ODD_{i,t} + Tar_{product\_i,t} \qquad (2)$$

Assuming the agent selects job i for processing at work-center j at time t, the corresponding reward function consists of four parts. The value of each part is normalized to ensure each part contributes equally to the overall reward function.

- $WD_{j,one-step-ahead\_j}$: is the sum of workload deviations of current work-center j and its one-step-ahead downstream work-center. Workload deviation is the difference between target workload and actual workload.
- $QT_{i,t}$: is the queue time of the job i, which is the time difference between current time t and job arrival time.
- $ODD_{i,t}$: is the difference between current time t and the operation due date for job i.
- $Tar_{product\_i,t}$: is the sum of tardiness of the jobs belonging to the same product type as job i in the whole wafer fabs. Tardiness is the deviation between current time t and final due date.

The first part, which is the sum of workload deviation of the current and downstream work-center, provides feedback on whether the actual workload is high or low compared to the target workload. As previously described, the agent considers the dynamic workload situation when learning the system. Therefore, the workload deviation tells the agent if selecting a job will have positive or negative effect on the workload situation, i.e., if it will result in an overloaded or underloaded work-center. The value of workload deviation can be positive or negative. From the viewpoint of workload balance, the agent should select a job to avoid congestion in downstream work-center, and consequently, the agent obtains a high value of workload deviation.

The queue time of the job is the second part of the reward. A high WIP level often occurs at a high-utilized work-center as jobs spend considerable time in the queue. In this case, queue time is a useful indicator that tells the agent to prioritize jobs with long queue time to reduce the WIP level. Additionally, this can potentially reduce the cycle time of the job.

The third part of the reward function is the deviation from the operation due date, which indicates whether the selected job will be delayed or not. The higher deviation value implies that the job is deviating more from its operation due date, and selecting such a job may have a cascading effect on subsequent operations. Therefore, the agent should choose jobs that minimize this deviation to avoid potential delays. Compared to the final due date, the operation due date is a more precise indictor for preventing

tardiness as it allows the agent to track the progress of each operation and take corrective actions in timely manner.

When a job is already tardy for its final due date, the last part of the reward function - accumulated tardiness of the product type becomes an important factor. Unlike the operation due date deviation, the accumulated tardiness is always positive and only considers jobs that are already late for their final due date. Thus, when more and more jobs of a product type become tardy, the agent is encouraged to prioritize processing jobs of that type to reduce the overall tardiness.

3 Simulation Model, Experiment and Results

3.1 Simulation Model

The whole wafer fabs dataset MIMAC6 from Measurement and Improvement of Manufacturing Capacities (MIMAC) is used for this study. The interested readers are referred to Fowler and Robinson (1995). The MIMAC6 is a typical complex wafer fabs model including:

- 9 products, 9 process flows, maximum 355 process steps.
- 24 wafers in a lot (job). 2777 lots are released per year under fab loading of 100%.
- 104 tool groups (work-centers), 228 tools (machines). 46 single processing tool groups, 58 batching processing tool groups.
- Sequence dependent setup, rework, MTTR (mean time to repair), and MTBF (mean time between failures) of tool group.

3.2 Simulation Experiment and Results

The concept of dispatch agent and deep-Q-network has been implemented in C++ and integrated into FX simulation software. Each work-center in the MIMAC6 model, which contains 104 work-centers, is theoretically controlled by a dispatch agent. Work-centers with high utilization rate are often overloaded, resulting in significant job queue time. To account for computational complexity, a preliminary study is conducted, and the deep reinforcement learning approach is applied to only the top 5 highest-utilized work-centers and their 27 upstream work-centers. As a result, a total of 32 work-centers are controlled by dispatch agents, each with its own deep-Q-network, while the remaining 72 work-centers are controlled by the FIFO rule. During the training phase, the MIMAC6 model is simulated for 104 weeks, while another simulation of 78 weeks is carried out during the deployment phase. The first 26 weeks are considered as warm-up periods and excluded from statistical analysis. The fab loading is set at 95%, and the target due date flow factor for operation due date and final due date is set at 2.2.

Figure 2 displays the evolution of WIP over 52 weeks. The curve for the FIFO rule shows the most fluctuation, while the ODD rule generates a smoother curve with less fluctuation. The MIVS and the composite rule are able to accelerate the movement of jobs, resulting in lower WIP curves that outperform the FIFO and ODD rules. Similarly, the DRL approach also produces a lower WIP curve as it balances WIP as part of its target. Moreover, starting from the 37th weeks onwards, the DRL approach significantly reduces the WIP level compared to the MIVS and composite rules.

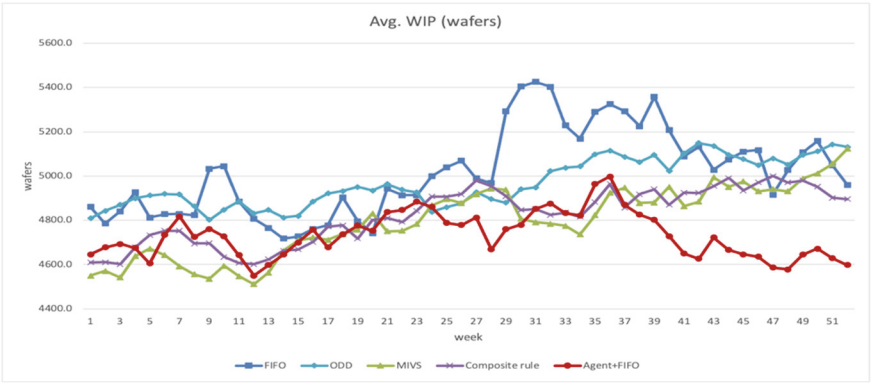

Figure 2: WIP curve comparison

Table 2 presents the performance of the whole wafer fabs, including the average cycle time, cycle time variance, cycle time upper percentile 95%, percent tardy job and average tardiness of tardy jobs. The DRL approach achieves the best performance in terms of average cycle time but not cycle time variance. It balances WIP to accelerate job movement while employing due date information to move jobs at a relatively smooth pace. As a result, 95% of job cycle times are lower than 35.7 days, which leads to the best tardiness performance compared to other four rules.

Table 2: Performance measurements comparison

Scenario	Avg. Cycle Time (days)	Cycle Time Variance (days^2)	Cycle Time Upper Percentile 95% (days)	Percent Tardy Job (%)	Avg. Tardiness for Tardy jobs (days)
FIFO	29.6	1.7	39.1	71.4	2.2
ODD	29.5	0.3	35.8	72.5	1.5
MIVS	28.5	1.3	37.2	56.3	1.7
Composite rule	28.3	1.0	36.3	53.2	0.9
DRL (Agent + FIFO)	27.2	0.8	35.7	40.6	0.7

4 Conclusion

This paper presented a dispatching study based on deep-Q-network reinforcement learning with the aim of achieving workload balance and on-time delivery simultaneously for wafer fabs. The dispatch agent with deep-Q-network is incorporated into FX simulation software. The simulation provides a training environment where the agent acquires knowledge and makes dispatch decisions.

When a machine becomes available, the agent takes the system states, such as the workload of the work-center and due date of the job, as input to train its own deep-Q-network. Then, the agent selects an action based on the output of the deep-Q-network. Finally, the agent obtains reward from the environment to improve its decision-making for future actions.

In contrast to earlier contributions, workload balance and on-time delivery are taken into consideration as objectives. Therefore, the system states are defined carefully to represent the status of the wafer fabs. In addition, in order to encourage the agent to select appropriate actions, the reward function is also properly formulated via the information of workload and tardiness. The simulation results indicate the DRL approach outperforms the FIFO, ODD, MIVS and composite rule in terms of the global performance of average cycle time and average tardiness. This indicates that the DRL approach is capable of achieving workload balance and on-time delivery simultaneously. For future work, we believe there is scope for further improvement when all work-centers are controlled by the agents learning decision via deep-Q-network.

References

Fowler, J.W.; Robinson, J.: Measurement and improvement of manufacturing capacities(MIMAC): final report. Technical Report 95062861A-TR, Sematech, Austin, 1995.

Keskincocak, P.; Tayur, S.: Due date management policies. In: Simchi-Levi D., Shen ZJ. (eds) Handbook of Quantitative Supply Chain Analysis. International Series in Operations Research & Management Science. Springer, Boston, MA, 2004, pp. 485-554.

Li, S.; Tang, T.; Collins, DW.: Minimum inventory variability schedule with application in semiconductor fabrication. IEEE Transactions on Semiconductor Manufacturing, February 1996, pp.145-149.

Panzer, M.; Bender, B.; Gronau, N.: Deep reinforcement learning in production planning and control: a systematic literature review. In: Herberger, D.; Hübner, M. (Eds.): Proceedings of the Conference on Production Systems and Logistics: CPSL 2021. Hannover: publish-Ing., 2021, pp.535-545.

Riedmiller, S.; Riedmiller, M.: A neural reinforcement learning approach to learn local dispatching policies in production scheduling. International Joint Conference on Artificial Intelligence 1999, pp.764-769.

Sakr, A.H.; Aboelhassan, A.; Yacout, S.; Bassetto, S.; Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems. Journal of Intelligent Manufacturing 34 (2021), pp. 1311-1324.

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

Zhou, Z.; Rose, O.: A composite rule combining due date control and wip balance in a wafer fab. In: S. Jain, R.R.Creasey, J. Himmelspach, K.P. White, and M. Fu (Eds.): Proceedings of the 2011 Winter Simulation Conference (WSC), Phoenix (USA), December 11th-14th December 2011, pp.2085-2092.

Zhou, Z.: WIP balance and due date control for complex job shops (wafer fabs). Ph.D. thesis, Department of Computer Science, die Universität der Bundeswehr München, Germany. http://athene-forschung.rz.unibw-muenchen.de/node?id=97064, accessed March 17, 2015.

Zhou, Z.; Rose, O.: A framework for effective shop floor control in wafer fabs. In: L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossett (Eds.): Proceedings of the 2015 Winter Simulation Conference (WSC), Huntington Beach (USA), December 6th-9th December 2015, pp.3001-3012.

Zhou, Z.; Rose, O.: A global wip oritented dispatching scheme: work-center workload balance without relying on target wip. In: N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son (Eds.): Proceedings of the 2019 Winter Simulation Conference (WSC), National Harbor (USA), December 8th-11th December 2019, pp.2212-2223.