Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

ON SCHEDULING A PHOTOLITHOGRAHY TOOLSET BASED ON A DEEP REINFORCEMENT LEARNING APPROACH WITH ACTION FILTER

Taehyung Kim
Hyeongook Kim
Tae-eog Lee

Department of Industrial and Systems Engineering
Korea Advanced Institute of Science and Technology
291 Daehak-ro, Yuseong-gu
Daejeon, 34141, SOUTH KOREA

James Robert Morrison

School of Engineering Technology
Central Michigan University
ET 130F, Mt Pleasant, MI 48859, USA

Eungjin Kim

Smart IT Team
Samsung Display Company
181 Samsung-ro, Tangjeon-myeon
Asan-si Chungcheongnam-Do, 31454, SOUTH KOREA

ABSTRACT

Production scheduling of semiconductor manufacturing tools is a challenging problem due to the complexity of the equipment and systems in modern wafer fabs. In our study, we focus on the photolithography toolset and consider it as a non-identical parallel machine scheduling problem with random lot arrivals and auxiliary resource constraints. The proposed methodology strives to learn a near optimal scheduling policy by incorporating WIP, masks, and the tardiness of jobs. An Action Filter (AF) is proposed as a methodology to eliminate illogical actions and speed the learning process of agents. The proposed model was evaluated in a simulation environment inspired by practical photolithography scheduling problems across various settings with reticle and qualification constraints. Our experiments demonstrated improved performance compared to typical rule-based strategies. Relative to our learning methods, weighted shortest processing time (WSPT) and apparent tardiness cost with setups (ATCS) rules perform 28% and 32% worse for weighted tardiness, respectively.

1 INTRODUCTION

In the semiconductor production process, the photolithography process is typically the fabricator bottleneck (Morrison and Martin 2007). In particular, since the production equipment for the photo process is expensive, purchase of additional resources to alleviate the bottleneck loading is limited. Therefore, production scheduling of the photolithography process is a key technology that increases the throughput and efficiency of the entire fab (Yoon and Lee 2004). However, due to the high complexity of the modern fab system, solving large-scale problems with an exact method is impossible.

We focus on scheduling the photolithography toolset and regard it as parallel machine scheduling with random arrivals and auxiliary constraints. It is known to be NP-hard (Lenstra et al. 1977). Furthermore, such a problem can be more complex when it has additional constraints such as machine eligibility.

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

Well known fab operators, such as Samsung Display Co., Ltd., utilize rule-based methods informed by the knowledge of field experts to schedule the machines. We consider rule-based scheduling as a method in which an expert person creates all dispatching rules, determines when they are active, and supports the creation of a processing method for each case. As the rule-based method allows the user to clearly grasp the operating principle, and operates well in most cases, it has been widely used and studied (Sha et al. 2006; Lee et al. 2003; Ching-Chin Chern and Yu-Lien Liu 2003). However, it is generally considered that no dispatching rule outperforms all other dispatching rules. Recently, Li et al. (2013) studied adaptive dispatching rules to address ongoing changes in fab environment and configuration.

Another important class of approaches is the use of formal heuristic methods, such as evolutionary algorithms or matheuristics. Such methods have been applied to solve scheduling problems in various settings with sequence-dependent setups (Ying and Cheng 2010), auxiliary constraints (Cakici and Mason 2007; Ham and Cho 2015) and eligibility constraints (Chen et al. 2016). Meanwhile, mixed integer programming(MIP) and constraints programming(CP) are an important optimization technique for semiconductor manufacturing industry. Such challenges which cannot be addressed in dispatching rules are highlighted in (Klemmt et al. 2017). Bixby et al. (2006) address real time dispatching paradigm including hot-lots, batch coordinating and Qtime restriction with CP. In (Ham et al. 2020), scheduling production and material transfer in photolithography area is investigated. Although solving problem with CP is not able to find feasible solution for medium-size instances, it is worth to mention that proposed approach find optimal solution in complex system which include jobs, machines, and vehicles.

Recently, studies that applied reinforcement learning to manufacturing process scheduling have been actively conducted by many researchers. Many of the papers on scheduling machines with reinforcement learning incorporate existing dispatching rules as RL agent actions (Aydin and Öztemel 2000; Wang and Usher 2005; Lee et al. 2019; Guo et al. 2020). Clearly, these approaches have an advantage as they utilize dispatching rules which contains domain-specific knowledge to exploit the learning ability of the agent. However, even such RL dispatch rule selection can be improved when considering highly constrained and complex scheduling problems, such as fab scheduling problems. The alternate method in Park et al. (2019) proposed selecting individual jobs as an action for wire and bonding production scheduling problems with multi-agent approach. Numerical studies showed that the proposed approach outperforms rule-based methods and heuristics. However, from the complexity of learning perspective, the dramatic increase in the size of the action space (from selecting a rule to selecting a lot) results in substantial additional computational complexity. Note that, although it is not related to the scheduling domain, Zahavy et al. (2018) shows that action elimination techniques can speed the learning process. To enable learning in fab size problems, motivated by their work, we apply the concept of action elimination to our problem informed by field-specific knowledge.

1.1 Contribution and organization

In this study, we develop

- an efficient Deep Q-Networks(DQN) algorithm for real-time sequential decision making in photolithography, and
- a domain specific AF to support the learning process by the elimination of illogical actions.

To reduce learning complexity, we allocate one agent to each individual machine. Our simulation experiments compare the performance with well-known existing dispatching rules for parallel machine scheduling problems. The performance improvement obtained by the proposed algorithm is reported.

The remainder of the paper is organized as follows. In Section 2, we formulate the problem as a parallel machine scheduling problem. Simulation modeling for the RL environment of the photolithography toolset and RL model is provided in Section 3. We assess the performance of the proposed model compared with the weighted shortest processing time (WSPT) rule and the apparent tardiness cost with setups (ATCS)

rule in Section 4. Concluding remarks, condensed results, and future research directions are provided in Section 5.

2 PROBLEM DESCRIPTION

Modern fabs consist of hundreds of tool groups, each of which conducts specific types of processes such as etching, deposition, etc. Each product requires that these processes be conducted multiple times in a re-entrant flow. In this paper, we focus solely on a photolithography toolset. Prior processes, subsequent processes, and the movement of lots are simplified as external arrivals. We model the arrival of jobs from the prior stages as independent random arrivals based on realistic fab data. Arrived jobs wait in an integrated buffer until an agent dispatches them to a machine. Figure 1 shows a overview of modeling the photolitography toolset centered scheduling problem.

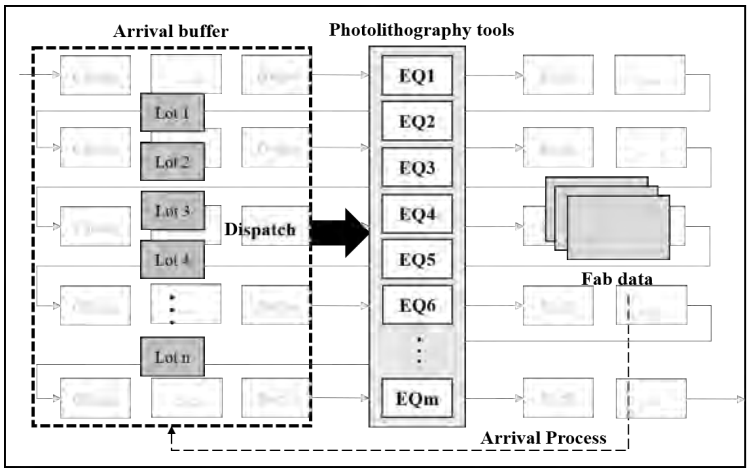


Figure 1: Overview of modeling photolithograhy tool scheduling problem.

We model our photolithography scheduling problem as a parallel machine scheduling problem with initial WIP, random arrivals, resource constraints (for reticles) and eligibility restrictions (lots may be allowed only on certain tools). There are m parallel machines (m_i) and n types of lots (LT_j). For job j, we denote the number of jobs that waiting for dispatching as Nw_j ($Nw_j \in \{1,2,,\infty\}$). Each LT_j is dedicated to process at specific machine($H_{j,i}$) and require reticle resource ($ret_{j,l}$, $l \in \{1,2,\ldots,r\}$). If prior LT_i and following LT_j requires different reticle resources($ret_{i,l} \neq ret_{j,l}$), the system need setup time (st) for change reticle. With $H_{j,i}$, each machine i can process certain types of lots ($n(i) = \sum_j H_{j,i}$). Each LT_j is assumed to arrived from formal steps with Poisson process (λ) and have duedate ($d_{j,k}$) for kth waiting job of LT_j. Unscheduled lots in integrated buffer wait until formal job at each machine to be processed. When a machine become idle, with $H_{j,i}$ and $ret_{j,l}$, agent separate out possible dispatching candidates.

3 PHOTOLITHOGRAHY SCHEDULER BY REINFORCEMENT LEARNING

In this section, we introduce the structure of the photolithography scheduler by reinforcement learning. Figure 2 shows the overall structure of proposed model. Arrival is generated from historical production data of fab. Each machine in a parallel toolset has an agent that dispatching unscheduled lots. The RL agent interacts with the integrated system simulation and observes system state and selects a lot to dispatch. The proposed model incorporates the machine eligibility constraints and reticle resource constraints. We first provide the simulation environment of a single machine in Section 3.1. In Section 3.2, the model of reinforcement learning, algorithm, state, reward, and action definition will be described.

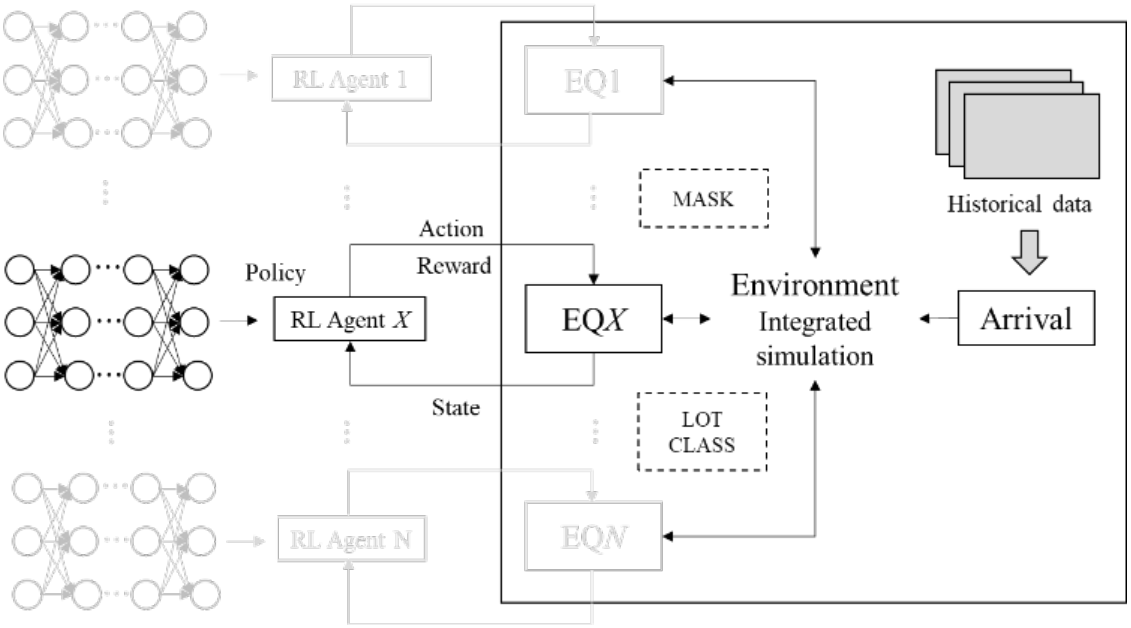


Figure 2: Overall framework of proposed RL architecture.

3.1 Simulation environment

Simulation for RL is modeled as process lots with the flow of events as shown in Figure 3. Figure 3 shows a simulation of one equipment, and the modularized simulations are gathered to compose a simulation centered on the photo process of the entire fab. This simulation was modeled with Discrete Event Simulation (DES). Basically, lots are continuously generated through *Arrive* event. The running equipment processes the lot of the buffer assigned to the equipment through the *Unload* event, and if there is no available mask for the equipment, it generates *Wait* event and waits until the mask becomes available.

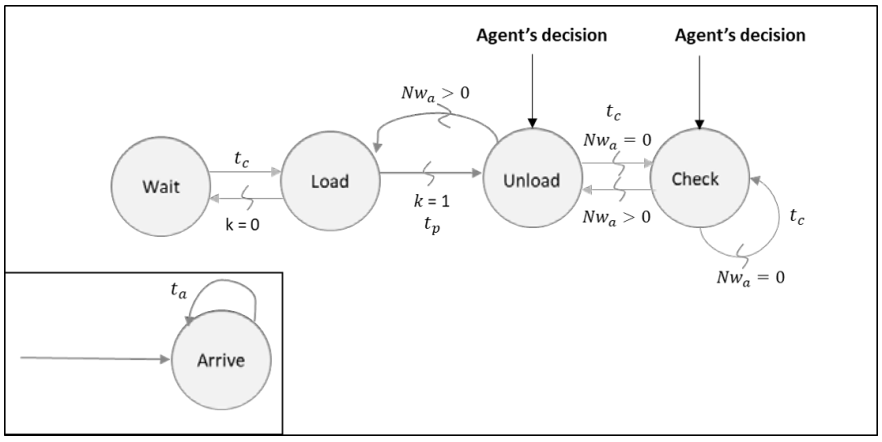


Figure 3: Simulation structure of single machine. t_a: interarrival time, t_c: processing time, Nw_a: number of jobs waiting of selected LT_a, k: mask availability and t_c: waiting time until next decision. Agent select an action when machine is idle, *Unload* event and *Check* event.

As above, the normal behavior of the lot and equipment is modeled through simulation, while a decision is made when the lot in the empty buffer space of the equipment that occurs whenever an *Unload* event occurs. Since there are lots that can be assigned several options among the lots waiting in the virtual arrival

buffer, it is necessary to decide to select an appropriate lot among them. At this time, the learning agent decides in consideration of WIP, mask use, and tightness to duedate caused by the preceding lot.

3.2 Reinforcement learning Model

For scheduling, this paper utilizes Deep-Q-Network (DQN). DQN is a Model-Free RL method that can generate near optimal scheduling without knowing the state transition in the specification of the Markov Decision Process (MDP). In the proposed fab environment, it is difficult to specify all the probabilistic processes of state transition. Furthermore, considering the dimension of the state of the model targeting the fab size, a deep reinforcement learning model that approximates the value by introducing an artificial neural network is an appropriate alternative. DQN is a representative methodology that approximates the value state function in the existing Q-learning using artificial neural networks. We shorten the general description of DQN, see (Mnih et al. 2013)

To employ DQN, the system must be formulated as RL problem. We propose the system state which represents the status of production, action, and reward in the following sections. Formulation of RL is constructed to incorporate domain knowledge which describes fundamental information for an agent to make a decision.

3.2.1 State description

The state s is designed to represent the status and variability of the production system. We define the following system state s for agent of machine i consist of nine state features.

Feature 1: Let Nw_j represents number of waiting jobs of lot type LT_j. Recall $n(i)$ is the number of lot types that can be processed in machine i, $n(i) = \sum_j H_{j,i}$. For $j \in \{1,2,\ldots,\text{n(i)}\}$, s_j^1 is defined as follows.

$$s_j^1 = \begin{cases} 0, & \text{if } Nw_j = 0, \\ Nw_j, & \text{if } Nw_j > 0. \end{cases}$$

Feature 2: s_l^2 represents the status of the mask. For $l \in \{1,2,\ldots,r\}$, s_l^2 is defined as follows.

$$s_l^2 = \begin{cases} 1, & \text{if mask l is available,} \\ 0, & \text{if mask l is occupied by machine.} \end{cases}$$

We define three features for tracking the tardiness of the jobs in the system. Through observing following features, an agent is capable of capturing not only the urgent job but also the overall tightness for waiting jobs of all lot types.

Feature 3: s_j^3 represents maximum tightness of waiting jobs for lot type LT_j. Recall Nw_j represent number of waiting jobs of LT_j. Let t the system time of the decision epoch. For $j \in \{1,2,\ldots,n(i)\}$, s_j^3 is defined as follows.

$$s_j^3 = \max_{k \in \{1,2,\ldots,Nw_j\}} (d_{j,k} - t). \tag{1}$$

Feature 4: s_j^4 represents minimum tightness of waiting jobs for lot type LT_j. For $j \in \{1,2,\ldots,n(i)\}$ s_j^4 is defined as follows.

$$s_j^4 = \min_{k \in \{1,2,\ldots,Nw_j\}} (d_{j,k} - t). \tag{2}$$

Feature 5: s_j^5 represents average tightness of waiting jobs for lot type LT_j. For $j \in \{1,2,\ldots,n(i)\}$ s_j^5 is defined as follows.

$$s_j^5 = \sum_{k=1}^{Nw_j} (d_{j,k} - t)/Nw_j. \tag{3}$$

We define four additional features for providing the information of urgency of jobs related to tightness. Only a specified definition of s_j^6 will be given, the others behave similarly.

Feature 6: s_j^6 represent the urgency of waiting jobs of LT_j. $p_{j,k}$ denote the processing time of lot type j at machine k. $E(p_{j,k})$ represents the expected processing time of lot type j at machine k. For $j \in \{1,2,\ldots,n(i)\}$, s_j^6 is defined as follows.

$$s_j^6 = \begin{cases} Nw_j, & \text{if } s_j^3 < E(p_{j,k}), \\ 0, & \text{otherwise.} \end{cases}$$

s_j^6 indicates most of waiting jobs in LT_j are tardy. Nw_j plays a role of the scaling factor that implies how many jobs in LT_j are tardy. This state feature gives direct information of tardiness compare to state features 3,4 and 5. For s_j^7, s_j^8, s_j^9 are equal to Nw_j if (3) $< E(p_{j,k}) \leq$ (1), (2) $< E(p_{j,k}) \leq$ (3), $E(p_{j,k}) <$ (2) respectively.

3.2.2 Action

At each decision point, the time at a machine become idle or at *Check* event, an agent select a lot in lot type LT_j and dispatch to machine. The action set a is a set of action a_i where each a_i represent the selection of lot in waiting lots. Note that the dimension of action set $N(a)$ is not equal to n. As mentioned on Section 2.1, the eligibility constraints for each machine eliminate the candidate and define the action set.

3.2.3 Reward

Let $T(t)$ be the set of tardy jobs in the system at time t and $\theta_t(j,k) = -I_{T(t)}(j,k)$. $I_{T(t)}$ is a indicator function where $I_{T(t)} = 1$ if lot k of LT_j is in $T(t)$, $I_{T(t)} = 0$, otherwise. Reward at decision point t define as following.

$$r(t) = \sum_{j=0}^{n} \sum_{k=0}^{Nw_j(t)} \int_{t_{o-1}}^{t_o} w_j \theta_t(j,k) dt \tag{4}$$

Equation (4) represents the length time of tardy jobs between t_o and t_{o-1}. The length of decision interval $t_o - t_{o-1}$ may different depend on the event of *Load* and *Check*. We resort to reward definition of (Zhang et al. 2007). In $P_m||w_jT_j$ scheduling problem, maximizing the sum of $r(t)$ over infinite time has proven to be equivalent to minimizing total weighted tardiness T.

3.2.4 Action Filter

The Action Filter(AF) proposed in this paper is a control method to increase the learning speed and performance of an agent. The AF is designed based on domain knowledge to prevent performing illogical action of an agent. In this domain, we utilize the WIP status and reticle status to prevent an agent selecting a lot that uses the exhausted resource.

We define two action filters for RL model of semiconductor process scheduling. First, the Lot Filter controls the action of arranging the lot when there is no waiting WIP of a specific lot type. Second, the Mask Filter controls the action of dispatching the lot when the mask required to produce a specific lot type is already occupied. Both filters use the characteristics of DQN's Value-based RL to eliminate action candidates and detailed descriptions are as follows.

Let us denote mask filter and WIP filters by $Mf = (mf_1, mf_2, \cdots, mf_n)$ and $Wf = (wf_1, wf_2, \cdots, wf_n$, respectively. From the above definition, we have the action filter (Af) is defined by component-wisely product of Mf and Wf, which means that

$$Af = (Af_1, Af_2, \cdots, Af_n) = (mf_1 \cdot wf_1, mf_2 \cdot wf_2, \cdots, mf_n \cdot wf_n). \tag{5}$$

Each i-th component of WIP filter, wf_i is given by $wf_i = 1 - \delta_{0,s_i^1}$. Here, we use the Kronecker's delta function δ_{xy}, which is 1 when $x = y$ and 0 otherwise. In addition, let us define g_{jl} as 1 when the mask l is available at LT_j, and 0 otherwise. Then, by definition each i-th component of mask filter, mf_i is given by $mf_i = \sum_l g_{il} s_l^2$. The State-Value function of DQN redefined by applying (5) is as follows.

$$Q_F(s,a;w) = Q(s,a;w) \circ Af.$$

For example, Q-value of LT_j with no WIP become temporarily zero when agent take action. Such actions are filtered out as an agent take greedy action except exploration.

4 COMPUTATIONAL EXPERIMENT

To measure the performance of the proposed algorithm, experiments are conducted in various settings. All experiment settings are set to aim for high-fab condition such that all machine's loading is greater than 90% as we model bottleneck photolithography toolsets. Eight different experiments are used to verify our attempt, parameters for each experiment are summarized in Table 1. For each experiment environment, inter-arrival time is generated with $exp(\lambda)$, processing time($p_{j,k}$) is generated with $U(0.9\delta,\ 1.1\delta)$, and weight of job(w_j) is generated with $U(0,1)$. We set the setup time st as 0.1δ for every experiments. Furthermore, due date of job($d_{j,k}$) is defined as $d_{j,k} = r_{j,k} + C \cdot p_{j,k}$ where $r_{j,k}$ is the release time of LT_j of lot k and C is the tightness factor which shows how arrived jobs are close to duedate. We set T as 2 in all test sets. Reticle constraints are generated as each reticle possess $\lfloor n/r \rfloor$ number of lots. Hyperparameter for training is shown in Table 2.

The proposed scheduling methodology is compared with the weighted shortest processing time (WSPT) and the apparent tardiness cost with setups (ATCS), which are traditional distribution rules used for parallel machine scheduling. With various machine settings, the robustness of the proposed algorithm was examined with respect to the size of the problem.

Table 1: Experiment settings for eight dataset.

Parameter	Value							
Experiment	1	2	3	4	5	6	7	8
m	6	8	10	12	14	16	18	20
n	80	100	120	140	160	180	200	220
k	15	20	25	30	35	40	45	50
λ	280	210	168	140	120	105	93.33	84
δ	1512							
Simulation horizon(s)	28800							

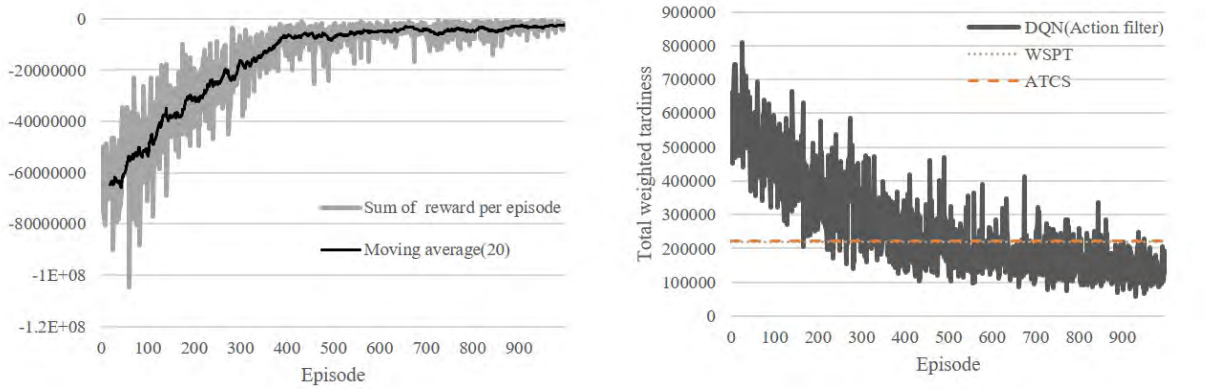
Table 2: Hyperparameters for reinforcement learning.

Hyperparmeter(RL)	Value
The number of hidden layers	5
Number of episode	1000
The number of nodes	2^8,2^9,2^{10},2^9,2^8
Learning rate	10^{-5}
Relay buffer size	500
Minibatch size	32
Discount factor	0.9
Target update frequency	100

We set the schedule horizon as eight hours which is a conventional working plan of fab operator known as one shift. We calculate tardiness for completed jobs during scheduling horizon $T = \max(0, C_{j,k} - d_{j,k})$ where $C_{j,k}$ is completion time of LT_j of lot k. In addition, $T_{remain} = \max(0, 28800 - d_{j,k})$ for the remaining jobs which has arrived during scheduling horizon but not processed. Mean processing time is fixed as 1512 and mean inter-arrival time varies according to the m to set machine loading greater than 90% in all test problems.

4.1 Training Result

Figure 4 shows total weighted tardiness and sum of reward per episodes in the training phase of the experiment set 4. As shown in Figure 4a, the sum of reward per episode is growing as agents learn dispatching policy. At beginning of training phase, agent try to explore the action even if there is no waiting job. With epsilon decay and AF, the sum of reward is growing until certain point of training phase. It was observed that, in every experiment, the sum of reward converged when the episode reaches over 500. It is also shown that the schedule with the lowest total weighted tardiness is obtained when training episodes exceed 500 as depicted in Figure 4b. The dashed lines in Figure 4b represent the average performance of WSPT and ATCS of 100 simulations.



(a) Training result of sum of reward per episode. (b) Training result of total weighted tardiness.

Figure 4: Training result of experiment4 with respect to number of episode.

The summary results are provided at Table 3. We evaluate the total weighted tardiness of the proposed model compare to WSPT and ATCS. The performance of DQN is evaluated with a trained network of 100 simulations. WSPT usually generates a better schedule with large number of machines and jobs compared to ATCS. DQN with Action Filter achieved 9%-64% lower total weighted tardiness in experiment sets. The result shows that the DQN with Action Filter learns flexible policy with the status of production and generates efficient schedule even in a fab-size environment. We also examine the DQN algorithm with same environment. However, without action filter, the training result showed that DQN is not able to learn dispatching policy. The learning curve diverged as the episode increase. The trained policy dispatch LT_j with no waiting job and continuously generate *Wait* events which deepen tardiness.

5 CONCLUSION

In this paper, we apply DQN with action filter to solve large-scale photolithography scheduling problem. Photolithography tools centered point of view, problem is modeled as $P_m|aux|\sum w_j T_j$. To reduce the complexity of learning for a realistic fab-size problem, we trained multi-agents for each machine which shares the information of production status. Each agent schedule a machine when the machine become

Table 3: Experiment result with total weight tardiness (relative performance index compared to DQN) of WSPT, ATCS and DQN(Action filter).

Experiment	WSPT	ATCS	DQN (Action filter)
1	154,942(1.41)	120,649(1.09)	110,001(1)
2	141,079(1.28)	144,342(1.31)	110,136(1)
3	145,812(1.17)	152,911(1.23)	124,108(1)
4	219,817(1.50)	222,432(1.52)	146,241(1)
5	222,271(1.27)	206,978(1.18)	174,021(1)
6	199,578(1.18)	216,424(1.28)	169,038(1)
7	223,136(1.27)	287,381(1.64)	174,619(1)
8	231,033(1.18)	273,471(1.32)	206,598(1)

idle with information of WIP, availability of reticle, and tightness of jobs. We define agent action as an individual lot type to explore broaden action candidates. Meanwhile, we suggest Action filter for photolithography scheduling problem which eliminate illogical action to prevent from inefficient exploration in the learning process. To assess the performance of the proposed model, eight different experiment environments are prepared. The experiment results shows that the proposed DQN algorithm with Action filter outperforms all other conventional dispatching rules in terms of total weighted tardiness.

Although training process for the fab-size problem takes a long time, the proposed model is capable of generating an efficient schedule with a trained Q-network within a short time. For the fab operator, once a train is finished, a rolling horizon strategy with a trained network would be a possible way to continuously update the production plan. Possible future research for the proposed model is increasing robustness with respect to the configuration of a fab. Fab-operators change their product mix and plan to cope with seasonal or temporary demand. However, the proposed model is subject to retrain when the product mix or fab configuration is changed. We are improving the state and action definition of the RL-model to overcome this issue.

ACKNOWLEDGEMENTS

This work was supported by Samsung Display Co., Ltd.

REFERENCES

Aydin, M. E., and E. Öztemel. 2000. "Dynamic job-shop scheduling using reinforcement learning agents". *Robotics and Autonomous Systems* 33(2-3):169–178.

Bixby, R., R. Burda, and D. Miller. 2006. "Short-interval detailed production scheduling in 300mm semiconductor manufacturing using mixed integer and constraint programming". In *The 17th Annual SEMI/IEEE ASMC 2006 Conference*, 148–154. Institute of Electrical and Electronics Engineers, Inc.

Cakici, E., and S. Mason. 2007. "Parallel machine scheduling subject to auxiliary resource constraints". *Production Planning and Control* 18(3):217–225.

Chen, J. C., Y.-Y. Chen, and Y. Liang. 2016. "Application of a genetic algorithm in solving the capacity allocation problem with machine dedication in the photolithography area". *Journal of Manufacturing Systems* 41:165–177.

Ching-Chin Chern, and Yu-Lien Liu. 2003. "Family-based scheduling rules of a sequence-dependent wafer fabrication system". *IEEE Transactions on Semiconductor Manufacturing* 16(1):15–25.

Guo, L., Z. Zhuang, Z. Huang, and W. Qin. 2020. "optimization of dynamic multi-objective non-identical parallel machine scheduling with multi-stage reinforcement learning". In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 1215–1219. Institute of Electrical and Electronics Engineers, Inc.

Ham, A., M.-J. Park, H.-J. Shin, S.-Y. Choi, and J. W. Fowler. 2020. "Integrated scheduling of jobs, reticles, machines, AMHS and ARHS in a semiconductor manufacturing". In *2020 Winter Simulation Conference (WSC)*, 1966–1973. Institute of Electrical and Electronics Engineers, Inc.

Ham, A. M., and M. Cho. 2015. "A practical two-phase approach to scheduling of photolithography production". *IEEE Transactions on Semiconductor Manufacturing* 28(3):367–373.

Klemmt, A., J. Kutschke, and C. Schubert. 2017. "From dispatching to scheduling: Challenges in integrating a generic optimization platform into semiconductor shop floor execution". In *2017 Winter Simulation Conference (WSC)*, 3691–3702. Institute of Electrical and Electronics Engineers, Inc.

Lee, G.-C., Y.-D. Kim, J.-G. Kim, and S.-H. Choi. 2003. "A dispatching rule-based approach to production scheduling in a printed circuit board manufacturing system". *Journal of the Operational Research Society* 54(10):1038–1049.

Lee, W.-J., B.-H. Kim, K. Ko, and H. Shin. 2019. "Simulation based multi-objective fab scheduling by using reinforcement learning". In *2019 Winter Simulation Conference (WSC)*, 2236–2247. Institute of Electrical and Electronics Engineers, Inc.

Lenstra, J. K., A. R. Kan, and P. Brucker. 1977. "Complexity of machine scheduling problems". In *Annals of discrete mathematics*, Volume 1, 343–362. Elsevier.

Li, L., Z. Sun, M. Zhou, and F. Qiao. 2013. "Adaptive Dispatching Rule for Semiconductor Wafer Fabrication Facility". *IEEE Transactions on Automation Science and Engineering* 10(2):354–364.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. "Playing atari with deep reinforcement learning". *arXiv preprint arXiv:1312.5602*.

Morrison, J. R., and D. P. Martin. 2007. "Performance evaluation of photolithography cluster tools". *OR spectrum* 29(3):375–389.

Park, I.-B., J. Huh, J. Kim, and J. Park. 2019. "A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities". *IEEE Transactions on Automation Science and Engineering* 17(3):1420–1431.

Sha, D., S. Hsu, Z. Che, and C. Chen. 2006. "A dispatching rule for photolithography scheduling with an on-line rework strategy". *Computers & Industrial Engineering* 50(3):233–247.

Wang, Y.-C., and J. M. Usher. 2005. "Application of reinforcement learning for agent-based production scheduling". *Engineering Applications of Artificial Intelligence* 18(1):73–82.

Ying, K.-C., and H.-M. Cheng. 2010. "Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic". *Expert Systems with Applications* 37(4):2848–2852.

Yoon, H. J., and D. Y. Lee. 2004. "Deadlock-free scheduling of photolithography equipment in semiconductor fabrication". *IEEE Transactions on Semiconductor Manufacturing* 17(1):42–54.

Zahavy, T., M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor. 2018. "Learn what not to learn: Action elimination with deep reinforcement learning". *arXiv preprint arXiv:1809.02121*.

Zhang, Z., L. Zheng, and M. X. Weng. 2007. "Dynamic parallel machine scheduling with mean weighted tardiness objective by Q-Learning". *The International Journal of Advanced Manufacturing Technology* 34(9-10):968–980.

AUTHOR BIOGRAPHIES

TAEHYUNG KIM is a Ph.D candidate in Department of Industrial and Systems Engineering, KAIST, South Korea. He hold B.S. degrees in Industrial and Systems Engineering, Buisness and Technology Management, and M.S. degree in Industrial and System Engineering from KAIST, South Korea. His email address is thkim@simlab.kaist.ac.kr.

HYUNGOOK KIM is a Ph.D candidate in Department of Industrial and Systems Engineering, KAIST, South Korea. He hold B.S. degrees in Industrial Engineering from Ajou University, South Korea, and M.S. degree in Industrial and System Engineering from KAIST, South Korea. His email address is hyung1405@kaist.ac.kr.

JAMES R. MORRISON received the Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign, USA in 2000. From 2008 – 2019, he was an Assistant/Associate Professor in the Department of Industrial and Systems Engineering, KAIST, Daejeon, South Korea. He is now Professor of Engineering and Technology at Central Michigan University, USA. His research focuses on smart production systems and UAV systems. His email address is morri1j@cmich.edu.

EUNJIN KIM received the Ph.D. degree in chemical engineering from KAIST, South Korea in 1998. After his degree, he has been working for Samsung Display Co. and has contributed himself in device process engineering, FAB automation and scheduling system. Focusing on artificial intelligence technology, he has been trying to apply it on manufacturing and human activities area. His email address is john.ej.kim@samsung.com.

TAE-EOG LEE received the Ph.D. degree in industrial and systems engineering from The Ohio State University, Columbus, OH, USA, in 1991. He is currently a Professor of Industrial and System Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His research interests include cyclic scheduling theory, scheduling and control theory of timed discrete-event dynamic systems, and their applications to automated manufacturing systems. His email address is telee@kaist.ac.kr.