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

A REINFORCEMENT LEARNING APPROACH FOR IMPROVED PHOTOLITHOGRAPHY SCHEDULES

Tao Zhang

Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Neubiberg, 85577, GERMANY

Kamil Erkan Kabak

Dept. of Industrial Engineering
Izmir University of Economics
Izmir, 35330, TURKEY

Cathal Heavey

CONFIRM Research Centre
School of Engineering
University of Limerick
Limerick, V94 T9PX, IRELAND

Oliver Rose

Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Neubiberg, 85577, GERMANY

ABSTRACT

A Reinforcement Learning (RL) model is applied for photolithography schedules with direct consideration of reentrant visits. The photolithography process is mainly regarded as a bottleneck process in semiconductor manufacturing, and improving its schedules would result in better performances. Most RL-based research do not consider revisits directly or guarantee convergence. A simplified discrete event simulation model of a fabrication facility is built, and a tabular Q-learning agent is embedded into the model to learn through scheduling. The learning environment considers states and actions consisting of information on reentrant flows. The agent dynamically chooses one rule from a pre-defined rule set to dispatch lots. The set includes the earliest stage first, the latest stage first, and 8 more composite rules. Finally, the proposed RL approach is compared with 7 single and 8 hybrid rules. The method presents a validated approach in terms of overall average cycle times.

1 INTRODUCTION

Reentrant Manufacturing Lines (RMLs) are considered as an example of complicated networks that have an extensive number of states, and hence finding exact solutions for them is difficult. Therefore, they are considered intractable or NP−hard (Ramirez-Hernandez and Fernandez 2007). These RMLs are also typical characteristics of semiconductor manufacturing, as each job visits a particular process a number of times in a repetitive manner that is also referred to as a cycle. Each visit of a job has a different operation, and, therefore, different types of jobs having different cycles compete for the same machine on a particular workstation or process. Additionally, the process may have additional constraints such as sequence-dependent setups and machine eligibilities that complicate the scheduling even further. For these reasons, proper management of job operation scheduling is crucial to improve production performance in semiconductor manufacturing.

In this study, the impact of dynamic selection of dispatching rules for the photolithography process is analyzed using a Reinforcement Learning (RL) approach and a simplified discrete event fabrication simulation model that explicitly takes into account RMLs. Due to the complexities of semiconductor

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

manufacturing, many similar simplified fab simulation models have previously been used to evaluate the overall fab performance (see Rose (2007)). In this study, the simulation modeling structure on these models is enhanced with the integration of an agent for sequential decision-making for photolithography operations in the simulation model. Stochasticity in process times along with machine failures is also embedded in the model. Although the popularity of machine learning applications has been increasing in recent years, most studies do not consider RMLs explicitly in fab schedules, and the role of RMLs on performance improvement over traditional scheduling rules is not specifically considered.

In the literature, RMLs have been used to control lot release and sequencing in 'Shop Floor Control' (Uzsoy et al. 1994). The control of RMLs has been investigated initially using simple models having two workcenters, three buffers, exponential process times, and a fixed number of revisits (see Liu et al. (2001)) and also by the adaptive versions by adding one more server and buffer (see Ramirez-Hernandez and Fernandez (2007)). However, such studies with simple models of RMLs do not consider production constraints such as sequence-dependent setups, machine capabilities and machine failures. To evaluate the role of RMLs, the RL approach proposed in this study considers the information on the number of revisits of lots visiting photolithography in addition to the current percentage of lots in the queue. Also, it compares the proposed RL approach with 10 customized rules that prioritize the lots according to their information on reentrant visits apart from 5 single rules. Thus, this study focuses on the impact of RMLs on the dynamic flexible job shop scheduling problem (DFSJP) with a proposed $Q-$learning algorithm to improve overall cycle time performance.

In a recent study, Tassel et al. (2023) applied RL and self-supervised learning to train a single RL agent's Neural Network (NN) acting as a global dispatcher for semiconductor manufacturing. The method outperforms the traditional hierarchical dispatching strategies typically used in semiconductor manufacturing plants, substantially reducing order tardiness and time until completion. However, in their study, information on reentrant flows is not taken directly into account in learning and scheduling. This information may have a great influence on scheduling within reentrant flows. Moreover, their definition of the state considers only features of lots and machines. This makes the agent lack a global view. Like Xie, S. et al. (2022) proposed, it would be more reasonable to consider the features of lots as the features of actions rather than the features of states. The global state of the whole fab also has an impact on the scheduling, for example, the current WIP level. In addition, the convergence of learning is very difficult to guarantee when the NN is used to approximate the policy or value function. The way in which they overcome this problem is not mentioned in their study. This study attempts to fill these gaps using the proposed RL approach and the distinct use of the RML information. This paper is organized as follows. The following section surveys the related literature in Section 2. The description of the problem is given in Section 3. This is followed by the introduction of the Reinforcement Learning (RL) model and the Q-learning algorithm in Sections 4 and 5. Section 6 presents related experiments and their results. Finally, the paper ends with discussion and conclusions in Section 7.

2 LITERATURE REVIEW

In the scheduling literature, rule-based approaches, also referred to as 'heuristic methods' or 'dispatch rules', are broadly applied by many studies (see Panwalker and Iskander (1977); Akcali et al. (2000)). However, as Min and Yih (2003) highlight that a well-performing rule does not exist or is not superior to the others for different production settings, particularly dynamically changing production environments. They apply different rules by associating four decision variables with multiobjectives by using a simulation model and competitive Neural Networks (NNs). They report their methodology as effective for a real-time control system and a preliminary control information saves time to obtain fast responses. However, such rules may suffer from lower quality solutions (Sarin et al. 2011). To overcome this issue, meta-heuristic approaches are also implemented to obtain higher quality solutions for flexible job-shop systems. Chen et al. (2016) use an application of Genetic Algorithm (GA) to minimize loading differences of machines in the photolithography area apart from a MIP model with machine capability and reticle constraints. Zhang et al.

(2018) propose improved imperialist competitive algorithm (ICA) for the schedules of photolithography machines using a rolling horizon strategy. However, these approaches might require significant computation time compared to rule-based approaches. On the other hand, recent machine learning-based scheduling approaches, particularly neural network (NN) based schedules, have been also widely applied in the literature. To illustrate, Arisha and Young (2004) develop a hybrid photolithography model that includes an integrated NN scheduler, and report significant improvements in tool utilization and lot cycle times.

Recently, the RL approach, a Markov Decision Process (MDP) based methodology that targets to maximize cumulative discounted rewards (Sutton and Barto 1998), has been broadly applied in semiconductor manufacturing to overcome computational or quality issues in solutions. With regard to the application of RL methods, Liu et al. (2001) use a Temporal Difference (TD) method, a class of incremental learning procedures, for a simplified reentrant system that includes one type of part and two service centers with one machine each. The TD method performs better than some known heuristics such as FBFS (First Buffer First Served), LBFS (Last Buffer Last Served), and UNWB (Workload Balancing Policy). Specifically, they observe close results to the WB policy on mean throughput, mean time before the first blocking, and mean number of parts before the first blocking. Q-learning that is a type of Model-free approach of RL (Watkins and Dayan 1992) is also applied for examining the RMLs. To illustrate, a variation of look-up table in Q-learning is applied by Ramirez-Hernandez and Fernandez (2005) considering the similar problem structure given by Liu et al. (2001). On another study, later they propose an approximate dynamic programming (ADP) algorithm for both control problems of job release and job sequencing in RMLs with the criterion of a discounted cost (DC) (Ramirez-Hernandez and Fernandez 2007). Using the TD learning with gradient-descent method, they apply a SARSA(λ) algorithm in their ADP approach. By the examining open RMLs, they present the comparative results of ADP approach by simulation experiments to numerical solutions of the modified policy iteration given in the literature. In addition to previous studies, they claim that their approach is also effective for improving the production performance for the short-term. However, as Park et al. (2020) point that Q-learning approaches may require extensive training time due to exponential growth of state and action space. To overcome this issue, then both Q-learning and NN-based methods (Gabel and Riedmiller 2008), deep RL and multiagent approaches (Park et al. 2020) are applied.

Regarding deep RL and FJSP, Waschneck et al. (2018) evaluate the impact of the application of deep Q-Network (DQN) agents together with a factory simulation model over standard dispatching heuristics to improve global scheduling performance in semiconductor manufacturing. Park et al. (2020) apply a new multi-agent RL approach to a Q-Network with a discrete event simulator to minimize the makespan for the die-attach and wire-bonding stages of a semiconductor packaging plant. Unlike previous studies, they apply a centralized approach which allows the share of fully connected NN by the agents to deal with changes particularly in the status of initial setups, number of machines, and production requirements. They compare their proposed method with a Genetic Algorithm (GA), a Two-Phase Deep Q-Network (TPDQN) by Waschneck et al. (2018) and several rule-based methods such as the Shortest Setup Time (SSU), Shortest Sum of Processing Time and Setup Time (SPTSSU), Most Remaining (MOR), Most Work Remaining (MWR) and Shortest Processing Time (SPT). They observe superior results to all of them with different datasets. On a non-identical parallel machine scheduling, Kim et al. (2021) apply two action filters that are WIP and mask filters to expedite and improve Q-learning with NNs by eliminating actions for photolithography toolsets. Their proposed method outperforms the Weighted Shortest Processing Time (WSPT) and the Apparent Tardiness Cost with Setups (ATCS) significantly for minimizing the weighted tardiness. In a recent study, Yedidsion et al. (2022) propose a deep RL method to find an optimal schedule under QTC (Queue-Time Constraints) with Proximal Policy Optimization (PPO) algorithm. The QTC is defined as a time violation for a lot to wait between two processes. A lot that violates such time constraints becomes scrapped or might need a rework. Thus, it affects the yield and throughput performance of the system. They compare their proposed RL method with 7 alternative agent methods named as kanban, capacity, frequency, random, always, never and Q-learning agents with makespan, number of queue-time violations, reward, cycle time, and utilization. Their PPO method presents results close to the best possible values when compared to those of the other methods. However, Q-learning is ranked as fourth in terms of cycle time after random and PPO methods in this study.

In summary, according to the RL and FJSP approaches, Q-learning with a single agent is one of the most traditional RL methods applied in the literature. Exponential growth of state and action spaces in Q-learning raises the issue of extensive training time. Various solutions such as multiagent or deep RL methods have recently been applied in the scheduling literature to improve both problem and computational performance. However, these approaches also have three main problems. The first is the difficulty of calculating the state features with existing RMLs. Second is the requirement of re-training of multiagents with changing manufacturing conditions, and third is the consideration of variability factors in the system (see Park et al. (2020), p. 1421). However, apart from these difficulties, Q-learning is also applied as a benchmark method for alternative approaches (see Yedidsion et al. (2022)). Accordingly, this study attempts to address the first issue by considering the features of lots as the features of actions and also using the information of RMLs. For this reason, the solution method is a modified variation of traditional Q-learning in order to test whether it overcomes the training issue. The definition of dispatch rules based on the updated information of lots can be attributed to another contribution of this study.

3 PROBLEM DESCRIPTION

The photolithography process is considered mainly a bottleneck process in semiconductor manufacturing since a different circuit layer is formed at each visit of a job by masking the circuit designs on silicon wafers (Kabak et al. 2013). Apart from being capital intensive and having a highly utilized equipment compared to other equipment, the process has additional constraints imposed by different capability constraints. Also, dynamic revisits of jobs into the process add further complexity. Figure 1 below denotes a schematic diagram of the simplified fab model with the photolithography process.

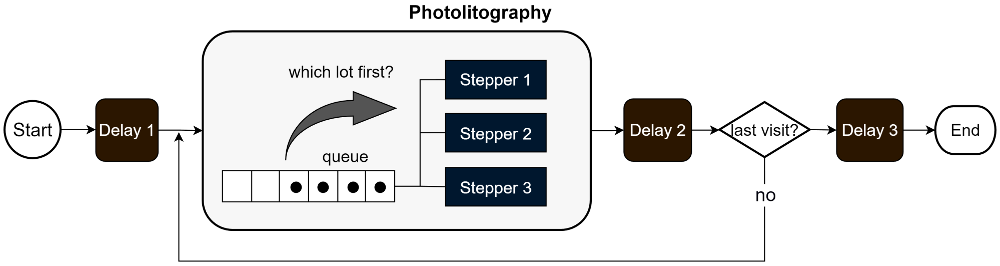

Figure 1: A simplified fab model with emphasis on photolithography.

According to Figure 1, the structure of the simulation model is based on three different types of delays. The first delay is assumed to encapsulate all the processing between lot starts in the system up to the first visit of the photolithography process. Similarly, the second delay is assumed to cover all the intermediate processing between each photolithography visit of a lot. Finally, the last delay considers all of the processing from the last photolithography visit from a lot to the end of production. In this problem, each job type is assumed to have a different technology and a sequence of recipe operations in the photolithography process. Each particular recipe operation has a machine eligibility constraint. That is, a particular set of toolsets is available for a particular recipe operation. This study considers a dynamic flexible job-shop scheduling problem (DFJSP) for the photolithography process with stochastic job arrivals. Dynamic selection of lots are evaluated as sequential decision process by the proposed RL approach since different types of jobs at different processing stages compete for the same machine in the photolithography workcentre. Stochastic arrival and processing times are considered with machine failures in the simulation model. The objective is to minimize overall cycle times by explicitly considering the status of RMLs at each lot assignment decision. Dynamic rule selection is a very common approach to real-time scheduling. The predefined rules are usually well-tested and perform well in certain situations. This study is trying to realize the approach by RL and find a good policy to do the selection of rules, i.e., using the right rule at the right time.

In the remaining sections, we are going to fill the research gap pointed out at the last of Section 1. Information about reentrant flows will be included in both the definition of state and action, and some global characteristics will also be added to the definition of the state (Section 4). To analyze the convergence, instead of NN, a tabular Q-learning is adopted to solve the problem (see Section 5). How the algorithm converges is discussed in Section 6 in an example model.

4 REINFORCEMENT LEARNING MODEL OF PHOTOLITHOGRAPHY SCHEDULING

Agents learn skills by interacting with the environment. Simply, the environment can be replaced by the simulation model. However, there is still a gap between the environment and the simulation. To bridge the gap, the state, action, and reward should be clearly defined and easily collected from the simulation (Xie et al. 2019). The probability that the current state changes to the next state after one action is also an important element in reinforcement learning. However, these probabilities are very hard to obtain from the manufacturing system (Zhang et al. 2017). When the agent learns using simulation, the simulation model actually finishes the transition between the states. The agent does not need to know transitions. The drawback is that only model-free algorithms can be used to solve unknown transition probabilities.

4.1 State Space

The state of a wafer fab is a combination of states of all entities in the system, like lots, machines, workers, etc. The state space will be quite huge if all these are included. So, aggregation among these trivial states is necessary, and the results can be the state of the environment. The following lists some possible aggregated features in state space: 1) Queue length/time; 2) Percentage of lots in the queue by product types; 3) Percentage of lots in the queue by current re-entrant times; 4) Current Work In Process (WIP) in fab. Since we focus only on photolithography, the first three features relate to the queue, with all steppers sharing a single queue. The queue length is the number of lots in the queue, and the queue time is the sum of the mean processing times of the lots in the queue. If the lots in the queue are grouped by product types, the percentage of each group can be a feature to demonstrate the product mix. If the lots are grouped by the current stage (currently visited counts) or a range of the current stage, the percentage of each group can also be one feature to reflect the progress of lots. The state of the whole fab may also influence scheduling, with the current WIP also being a feature.

4.2 Action Set

Lots of RL algorithms are only suitable for the fixed action set. If we consider jobs as actions in our study, we cannot fix the number of actions. We must use features of actions. The algorithms cannot be used directly then. As mentioned before, a single rule cannot perform well in all situations. The best rule should be selected dynamically according to the actual state of the wafer fab. In this study, the agent will select rules from a predefined set of rules. The action set is fixed. Most algorithms can be used directly without any modifications. The rules, stated below, are put into the set and considered as the action set. Especially, two re-entrant-related single rules are proposed: the earliest stage first and the latest stage first.

- First in first out (FIFO)
- Last in first out (LIFO)
- Shortest processing time first (SPT)
- Longest processing time first (LPT)
- Random
- Earliest stage (ES)
- Latest stage(LS)
- Earliest stage + FIFO (ESFIFO)
- Earliest stage + LIFO (ESLIFO)
- Earliest stage + SPT (ESSPT)
- Earliest stage + SPT (ESLPT)
- Latest stage + FIFO (LSFIFO)
- Latest stage + LIFO (LSLIFO)
- Latest stage + SPT (LSSPT)
- Latest stage + LPT (LSLPT)

The first arriving lot is selected first in FIFO, and the lot with the shortest processing time is selected first in SPT. LIFO and LPT are reverses of FIFO and SPT. The random rule gives lots random priorities. The earliest stage rule grants jobs with a smaller reentrant counter higher priorities, by contrast, the latest state rule grants jobs with bigger counter lower priorities. The others are compound rules. If two rules are combined to form one rule, the 1^{st} level rule selects the lots first, and the 2^{nd} level rule selects the lots if more lots have the same priorities as the 1^{st}level. The random rule is used as the last hidden level of the other rules in this study.

4.3 Reward

The goal of RL is to maximize the cumulative reward. This is different from the overall goal of the scheduling. Therefore, the definition of the reward must ensure that the goal of scheduling is automatically achieved while the cumulative reward is maximized (Xie et al. 2023). Because the objective is to minimize the average cycle time, this is equivalent to minimizing the WIP. This stance could be verified by the Little's Law for the stable systems. Also, WIP causes additional inventory costs in the system. Therefore, the reward can be a function of the cost of the inventory. To obtain the cost, first the average level of WIP, x_n, between the decision steps $n-1$ and n is calculated in the following.

$$x_n = \frac{1}{T_n}\sum_{i=1}^{C_n} y_i t_i$$

where C_n is the number of WIP changes between step $n-1$ and n, t is the duration between two changes in the WIP, and y_i is WIP level at i. Thus, $\sum_{i=1}^{C_n} t_i = T_n$, where T_n is the time between step $n-1$ and n. The inventory cost o_n can be calculated by

$$o_n = w x_n T_n = w\sum_{i=1}^{C_n} y_i t_i$$

where w is the inventory price per time per lot. The lower the cost, the more reward the agent receives. So, the immediate reward r is considered the inverse of the cost. Since the inventory price w is always kept the same, it can be omitted from the equation.

$$r = -o_n = -(w\sum_{i=1}^{C_n} y_i t_i) \approx -\sum_{i=1}^{C_n} y_i t_i$$

5 REINFORCEMENT LEARNING IN WAFER FAB SIMULATION

In the previous section, the RL model is defined. This section will address the approach to solving the RL model. Naturally, it is impractical for agents to interact with the real manufacturing system to learn scheduling skills because the real system cannot follow instructions from the agents without the validation of the agents, and the validation can only be performed after learning. Thus, the simulation model of the manufacturing system replaces the real system. After being trained and validated, the agent can directly/indirectly connect to the manufacturing execution system (MES) to dispatch lots.

5.1 Interaction between RL and Simulation

Reinforcement learning is embedded in a simulation experiment in which the simulator performs many replications. Each replication is one learning epoch. The experiment ends when the epoch reaches the maximum. The experiment can be for training or evaluation. Before starting a training experiment, the featured state space and the predefined rule set are created and used to build the agent. The trained agent is saved to local files at the end of the experiment. When an evaluation experiment starts, the agent is loaded and created from local files.

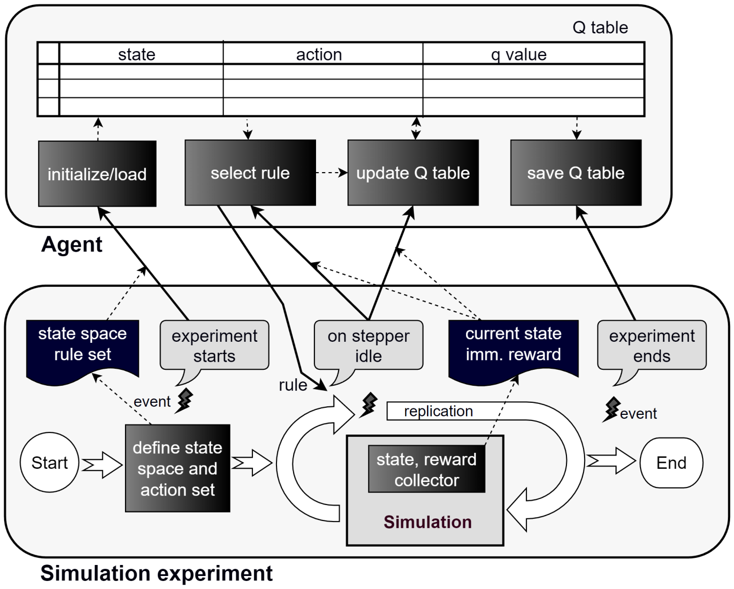

Figure 2: Tabular Q learning agent with simulation experiment.

The simulator communicates with the agent whenever a stepper becomes idle and the queue length is greater than 1. In the training experiment, the simulator collects and sends the current state and reward to the agent and asks the agent for a rule. The answer to the rule is used to grant the lots priorities which will be used by the simulator to take the lot, i.e., the one with the highest priority. In the evaluation experiment, the simulator only sends the current state to the agent without the reward. More details are shown in Figure 2 with a tabular Q-learning agent that is used in this study.

5.2 Tabular Q-learning Algorithm for Photolithography Scheduling

The tabular Q-learning algorithm is a very classical model-free algorithm for reinforcement learning. The Q values of all state-action pairs are stored and updated in a table (Sutton and Barto 1998). To create the table, several levels are defined for each state feature. The number of states is the size of the complete combination of levels in all features. When the agent is initialized in the simulator, it creates the queue table with a size of $|S| * |A|$, where S is the state space and A is the set of actions/rules. Whenever the simulator asks for a rule, a method

```
 def selectRule(state,reward)
```

in the agent is called. The rule is selected by a $\varepsilon - greedy$ algorithm with respect to the Q table. This guarantees the exploration of the state space while exploitation of the gained experiences. The Q value of the previous state-action pair is updated in the current step (1 step look ahead). A piece of code in the algorithm is shown below. Note that the method lrDecay(α) returns the decayed learning rate from the current decision step, where α is the initial learning rate. This helps the algorithm to converge. γ denotes the discount factor.

6 EXPERIMENTS AND RESULTS

The simplified fab model is built in Anylogic. Experiments are carried out on the simulation model to investigate the convergence of the algorithm and the performance in the simulation compared with the single rules in the rule set.

Algorithm 1 Behaviours of Q-learning agent

```
def init($S$, $A$)
    $Q \leftarrow Matrix(|S|,|A|)$ with random values
    $s' \leftarrow null$, $\varepsilon \leftarrow 0.1$, $\gamma \leftarrow 0.9$ ,$\alpha \leftarrow 0.1$, $training \leftarrow True$
def Q($s$, $a$) return Q[$s$][$a$]
def selectRule ($s$, $r$)
    $rnd \leftarrow$ random number
    if $training$ and $rnd < \varepsilon$
        select rule $a$ randomly from $A$
    else
        $a \leftarrow argmax_{a \in A}[Q(\text{s,a})]$
    if $training$ and $s' \neq null$
        $Q(s',a') \leftarrow Q(s',a') + lrDecay(\alpha) \times (r + \gamma \times max_{a \in A}[Q(s,a)] - Q(s',a'))$
        $s' \leftarrow s$, $a' \leftarrow a$
    return $a$
```

6.1 Experiment Environment

In the simulation model, the fab produces 3 types of products (A, B, and C) with 5 identical steppers in the photolithography workcenter. The interarrival time of lots into the fab follows an exponential distribution with a mean of 192. The products are mixed equally. All processing times are normally distributed and are manually adjusted to ensure 85% utilization on the steppers. The mean processing times of the products (A, B, and C) in the steppes are 55, 35, and 20, respectively. The mean times at Delay 1 and Delay 3 are independent of product types and are 20 and 50, respectively. However, the mean times at Delay 2 are product type dependent and they have means of 20, 15 and 30 for products A, B and C. All products visit the steppers 20 times. The maintenance is performed on the steppers in a uniform distributed interval of from 200 to 240 with also unifor distributed time from 10 to 24. The steppers break down randomly. The time to next failure follows the TriangularAV distribution with a mean of 1000 and a variability of 0.1. The time to repair also follows the same distribution with a mean of 10 and a variability of 0.1. In the RL model, the following seven features with certain levels form the state space: 1)Queue length; 2) Percentage of product A in the queue; 3) Percentage of product B in the queue; 4) Percentage of lots between stages 1-5; 5) Percentage of lots between stages 6-10; 6) Percentage of lots between stages 11-15; 7) Percentage of lots between stages 16-20. All percentage-related features have 2 levels: 0.0-0.5 and 0.5-1.0. The queue length remains within 5 levels: 0-2, 3-5, 6-8, 9-11, and 12-∞. So, the size of the state space is 320. 15 rules form the action set. As a result, the length of the Q table is 4800. The parameters of the learning algorithm remain the same as the code shown in Section 5.2.

6.2 Convergence of Q-Learning

Since the simulation environment has not a terminating state, the simulation ends after a fixed duration of two years. The experiment runs for 20,000 replications, i.e., 20,000 learning epochs for the agent. The learning rate decays during the experiment. The total reward and the average cycle time in each replication are collected during the experiment and with the replication index as the x-axis shown in Figure 3.

Based on the density of dots, the plots show the probability that the agent produces better results in both the total reward and the average cycle times. Due to the reducing learning rate, the range also becomes narrower. However, because the soft policy, i.e., $\varepsilon - greedy$, always explores the state space with probability ε. It can cause very bad or good results in that replication. Even with a very low probability, once a bad move occurs, it can destroy the whole replication because the follow-up situations in decision-making may change too. This is why the rewards and cycle times do not converge in a very narrow range. Of course, ε

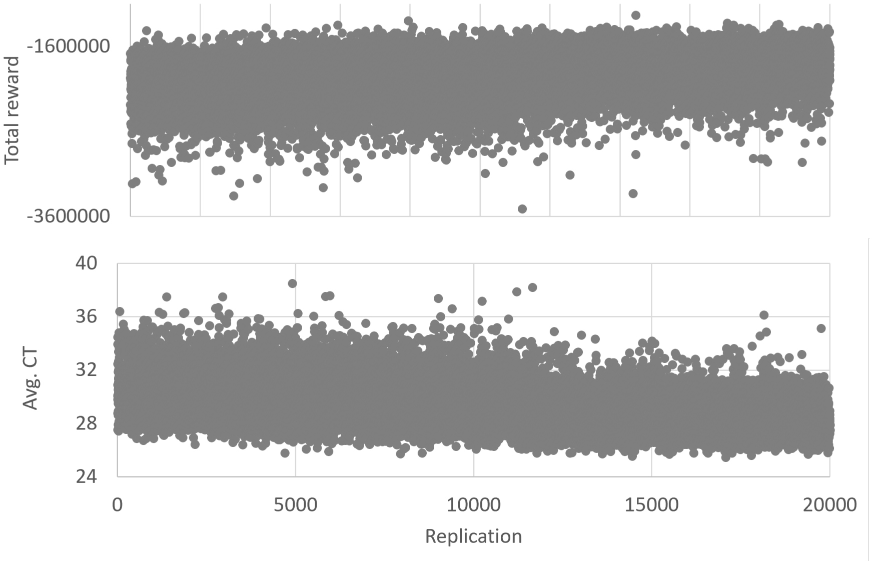

Figure 3: Total reward and average cycle times over replications.

can be decayed over the learning. Since the learning rate is also decayed, the collaboration between two decays may greatly influence the convergence. The study about this will be carried out in future work.

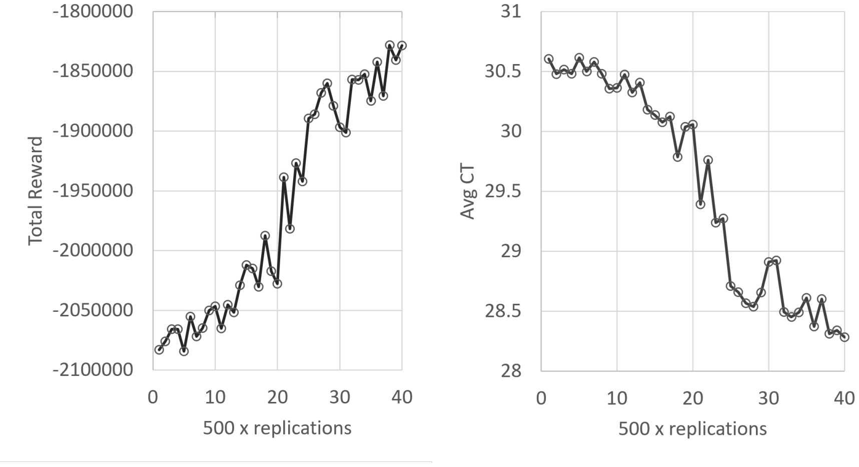

Figure 4: Total reward and average cycle times over every 500 replications.

Looking at Figure 3, it is very difficult to see the convergence. If every 500 successive replications is considered as one period, then Figure 4 shows the average of the total reward and cycle times. This figure clearly shows the convergence of the results. The total reward received from each replication is increasing while the average cycle time is decreasing.

6.3 Comparison with Other Decision Rules

The trained agent (RL) is compared with the single rules in the rule set. For each rule and the agent, the simulation runs 100 times and the outputs of the average cycle time are plotted (see Figure 5). The best rule is the stage-related rule ESSPT (26.99). This proves that a better decision is made by involving the information of reentrant flows. ESFIFO, ESLIFO, ESSPT, ESLPT, and ES always outperform the single rule without ES. All the earliest stage rules have very good performance while the latest stage rules produce very long average cycle times. Agent performance (RL, 28.98) is not the best but is very close to the best results. The reason why simple rules outperform the agent is due to the heavily simplified fab model. The simple rules usually work better in simple systems. The agent is very likely to perform better in more complicated models. More details about the experiment results can be found in Table 1 and 2.

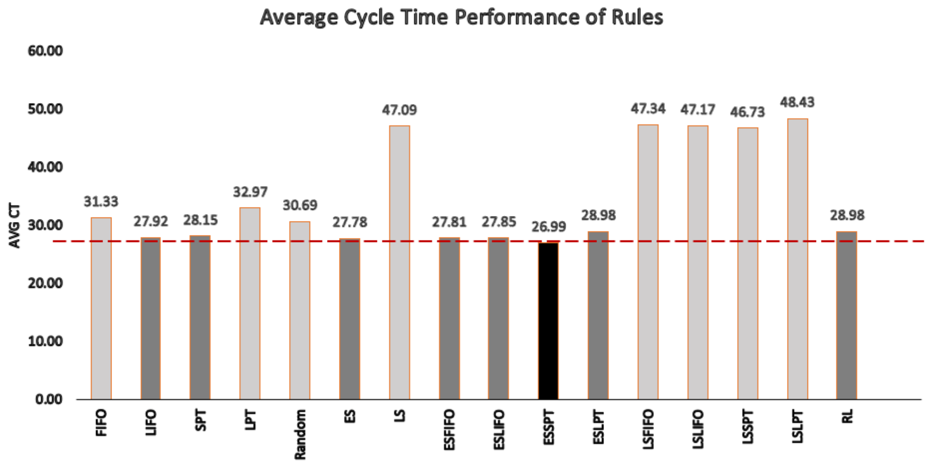

Figure 5: Comparison with other rules.

Table 1: Cycle time performance of single dispatch rules.

Rule	Mean ± Half-Width	Min	Max	Std Dev
FIFO	31.33 ± 0.15	26.04	50.79	2.35
LIFO	27.92 ± 0.10	24.63	46.69	1.61
SPT	28.15 ± 0.10	24.58	39.43	1.63
LPT	32.97 ± 0.17	26.59	48.80	2.78
ES*	27.78 ± 0.09	24.50	36.73	1.51
LS	47.09 ± 0.42	33.60	79.70	6.74
Random	30.69 ± 0.14	26.38	39.95	2.18
RL	28.98 ± 0.14	25.07	42.96	2.34

7 DISCUSSION AND CONCLUSIONS

In this study, the performance of the RL approach to scheduling in RMLs is analyzed with a simplified fab simulation model focusing on photolithography. According to the results of simulation experiments, the reentrant situation of lots influences the scheduling, however, the role of the RMLs is not considered in most scheduling studies. Moreover, NNs are often used to approximate policies and value functions, but they do not guarantee of convergence. Tabular-based algorithms can guarantee convergence and can be also used to solve problems having very big state space as long as enough computation power supports them. If it is impossible to set up corresponding hardware, anyway, the state space can be reduced by some dimension reduction algorithms or domain knowledge first. For this reason, an agent with a Q-learning algorithm is embedded into the simulation model and a set of composite dispatching rules along with single rules is defined as the set of actions. It is noted that the composite dispatching rules include the status of reentrant visits. Accordingly, the agent picks a particular rule dynamically for sequential photolithography decisions based on reward values. The results from the experiments confirm that the Q-learning approach can be used to perform scheduling in RMLs in the simplified model. Future research can include the evaluation of the

Table 2: Cycle time performance of composite dispatch rules.

Rule	Mean ± Half-Width	Min	Max	Std Dev
ESFIFO	27.81 ± 0.09	24.62	34.35	1.49
ESLIFO	27.85 ± 0.09	24.60	35.21	1.50
ESSPT*	26.99 ± 0.08	24.22	33.63	1.21
ESLPT	28.98 ± 0.12	24.98	39.09	1.94
LSFIFO	47.34 ± 0.43	32.47	85.79	6.99
LSLIFO	47.17 ± 0.49	32.33	101.12	7.92
LSSPT	46.73 ± 0.45	32.90	89.67	7.33
LSLPT	48.43 ± 0.47	33.14	84.59	7.65
RL	28.98 ± 0.14	25.07	42.96	2.34

impact of RMLs with a more detailed simulation model that takes into account additional main processes such as wet benches, furnaces, implant, and their production constraints. In addition, the proposed approach could be compared to deep RL approaches, and results could be validated by the different datasets obtained from real-time fab data.

REFERENCES

Akcali, E., K. Nemoto, and R. Uzsoy. 2000. “Alternative Loading and Dispatching Policies for Furnace Operations in Semiconductor Manufacturing: a Comparison by Simulation”. In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1428–1435. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Arisha, A., and P. Young. 2004. “Intelligent Simulation-based Lot Scheduling of Photolithography Toolsets in a Wafer Fabrication Facility”. In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1935–1942. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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.

Gabel, T., and M. Riedmiller. 2008. “Adaptive Reactive Job-shop Scheduling with Reinforcement Learning Agents”. *International Journal of Information Technology and Intelligent Computing* 24(4):1–60.

Kabak, K. E., P. C. Heavey, and V.Corbett. 2013. “Impact of Tool Recipe Constraints on the Photolithography Area in an ASIC Fabrication Environment”. *IEEE Transactions on Semiconductor Manufacturing* 26(1):53–68.

Kim, T., H. Kim, T.-e. Lee, J. R. Morrison, and E. Kim. 2021. “On Scheduling a Photolithograhy Toolset Based on a Deep Reinforcement Learning Approach with Action Filter”. In *Proceedings of the 2021 Winter Simulation Conference*, edited by Sojung Kim, Ben Feng, Katy Smith, Sara Masoud, Zeyu Zheng, Claudia Szabo, and Margaret Loper, 1–10. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Liu, C., H. Jin, Y. Tian, and H. Yu. 2001. “Reinforcement Learning Approach to Re-entrant Manufacturing System Scheduling”. In *Proceedings of the 2001 International Conferences on Info-Tech and Info-Net*, edited by X.Y. Zhong, Junfeng Shi, Xia Lin, Volume 3, 280–285. Beiing: Institute of Electrical and Electronics Engineers, Inc.

Min, H.-S., and Y. Yih. 2003. “Selection of Dispatching Rules on Multiple Dispatching Decision Points in Real-time Scheduling of a Semiconductor Wafer Fabrication System”. *International Journal of Production Research* 41(16):3921–3941.

Panwalker, S., and W. Iskander. 1977. “A Survey of Scheduling Rules”. *Operations Research* 25:45–61.

Park, I.-B., J. Huh, J. Kim, and J. Park. 2020. “A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities”. *IEEE Transactions on Automation Science and Engineering* 17(3):1420–1431.

Ramirez-Hernandez, J. A., and E. Fernandez. 2005. “A Case Study in Scheduling Reentrant Manufacturing Lines: Optimal and Simulation-based Approaches”. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, 2158–2163. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ramirez-Hernandez, J. A., and E. Fernandez. 2007. “An Approximate Dynamic Programming Approach for Job Releasing and Sequencing in a Reentrant Manufacturing Line”. In *Proceedings of the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, edited by Derong Liu, Remi Munos, Jennie Si, Donald C. Wunsch, 201–208. New York, N.Y.: Institute of Electrical and Electronics Engineers, Inc.

Rose, O. 2007. “Improved Simple Simulation Models for Semiconductor Wafer Factories”. In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1708–1712. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sarin, S. C., A. Varadarajan, and L. Wang. 2011. “A Survey of Dispatching Rules for Operational Control in Wafer Fabrication”. *Production Planning & Control* 22(1):4–24.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement Learning: An Introduction*. 1st ed, Volume 1. Cambridge, MA, USA: MIT Press.

Tassel, P., B. Kovács, M. Gebser, K. Schekotihin, P. Stöckermann, and G. Seidel. 2023. “Semiconductor Fab Scheduling with Self-Supervised and Reinforcement Learning”. https://arxiv.org/abs/2302.07162, accessed 15.08.2023.

Uzsoy, R., C. Lee, and L. A. Martin-Vega. 1994. “A Review of Production Planning and Scheduling Models in the Semiconductor Industry Part II: Shop-floor Control”. *IIE Transactions* 26(6):44–55.

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 29th Annual SEMI Advanced Semiconductor Manufacturing Conference*, 301–306. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Watkins, C. J. C. H., and P. Dayan. 1992. “Q-learning”. *Machine Learning* 8(3-4):53–68.

Xie, S., T. Zhang, and O. Rose. 2019. “Online Single Machine Scheduling Based on Simulation and Reinforcement Learning”. In *Proceedings of the 2019 ASIM Simulation in Produktion und Logistik*, edited by Matthias Putz and Andreas Schlegel, 59–68. Auerbach: Wissenschaftliche Scripten.

Xie, S., T. Zhang, and O. Rose. 2023. “Reward Calculation in Real-time Scheduling Based on Simulation and Q-learning”. In *Proceedings of the 2023 EUROSIM Congress*. July 3nd-5th, Amsterdam, Netherlands.

Xie, S., T. Zhang, and O. Rose. 2022. “Real-Time Scheduling Based on Simulation and Deep Reinforcement Learning with Featured Action Space”. In *Proceedings of the 2022 Winter Simulation Conference*, edited by Ben Feng, Giulia Pedrielli, Yijie Peng, Sara Shashaani, Eunhye Song, Canan Gunes Corlu, Loo Hay Lee, Ek Peng Chew, Theresa Roeder, and Peter Lendermann, 1731–1739. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Yedidsion, H., P. Dawadi, D. Norman, and E. Zarifoglu. 2022. “Deep Reinforcement Learning for Queue-Time Management in Semiconductor Manufacturing”. In *Proceedings of the 2022 Winter Simulation Conference*, edited by Ben Feng, Giulia Pedrielli, Yijie Peng, Sara Shashaani, Eunhye Song, Canan Gunes Corlu, Loo Hay Lee, Ek Peng Chew, Theresa Roeder, and Peter Lendermann, 3275–3284. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhang, P., Y. Lv, and J. Zhang. 2018. “An Improved Imperialist Competitive Algorithm Based Photolithography Machines Scheduling”. *International Journal of Production Research* 56(3):1017–1029.

Zhang, T., S. Xie, and O. Rose. 2017. “Real-time Job Shop Scheduling Based on Simulation and Markov Decision Processes”. In *Proceedings of the 2017 Winter Simulation Conference*, edited by Victor W.K. Chan, Andrea D’Ambrogio, Gregory Zacharewicz, Navonil Mustafee, Gabriel Wainer, and Ernest H. Page, 3899–3907. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

TAO ZHANG is a Research Assistant at the Universität der Bundeswehr München, and he holds an M.S. in Metallurgical Engineering from Chongqing University, China, and a Ph.D. in Computer Science from the Universität der Bundeswehr München, Germany. His research interest is working on production planning and scheduling, the main focus of his research is on the modeling and simulation of complex systems and intelligent optimization algorithms. His email address is tao.zhang@unibw.de.

KAMIL ERKAN KABAK is an Assistant Professor in the Department of Industrial Engineering, Izmir University of Economics. He received the Bachelor’s degree from the Middle East Technical University in Ankara, Turkey, the Master’s degree from the Department of Industrial Engineering, Dokuz Eylul University, Izmir, Turkey, and the Ph.D. degree from the Department of Design and Manufacturing Technology, University of Limerick, Limerick, Ireland. His research interests include simulation modeling and its application to complex manufacturing systems, stochastic processes and machine learning. His email address is: erkan.Kabak@izmirekonomi.edu.tr.

CATHAL HEAVEY is an Associate Professor in the School of Engineering at the University of Limerick. He is an Industrial Engineering graduate of the National University of Ireland (University College Galway) and holds an M. Eng.Sc. and Ph.D. from the same University. He has published in the areas of queuing and simulation modeling. His research interests include simulation modeling of discrete-event systems; modeling and analysis of supply chains and manufacturing systems; process modeling; and decision support systems. His email address is cathal.heavey@ul.ie.

OLIVER ROSE holds the Chair for Modeling and Simulation at the Department of Computer Science of the Universität der Bundeswehr, Germany. He received a M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of INFORMS Simulation Society, ASIM, and GI. His email address is oliver.rose@unibw.de.