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

MACHINE LEARNING-BASED PERIODIC SETUP CHANGES FOR SEMICONDUCTOR MANUFACTURING MACHINES

Je-Hun Lee
Hyun-Jung Kim*

Department of Industrial and Systems Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak Street
Daejeon, 34141, REPUBLIC OF KOERA

Young Kim
Yun Bae Kim

Department of Industrial Engineering
Sungkyunkwan University
2066 Seo-bu Street
Suwon, 16419, REPUBLIC OF KOERA

Byung-Hee Kim
Gu-Hwan Chung

VMS Solutions Co., Ltd.
U-Tower Building A
767 Sinsu Street
Yongin, 16827, REPUBLIC OF KOERA

ABSTRACT

Semiconductor manufacturing machines, especially for photo-lithography processes, require large setup times when changing job types. Hence, setup operations do not often occur unless there is no job to be processed. In practice, a simulation-based method that predicts the incoming WIP is often used to determine whether changing machine setup states or not. The simulation-based method can provide useful information on the future production environment with a high accuracy but takes a long time, which can delay the setup change decisions. Therefore, this work proposes a machine learning-based approach that determines setup states of the machines. The proposed method shows better performance than several heuristic rules in terms of movement.

1 INTRODUCTION

Semiconductor manufacturing processes can be considered to be hybrid flow or job shops with reentrant flows where there are multiple machines in each stage, and jobs are sequentially processed on the stages and visit some stages several times. The manufacturing machines, especially for photo-lithography processes, require large setup times when changing job types. Therefore, setup operations do not often occur and are performed when necessary depending on the production environment. Some processing machines with large setup times mainly use a periodic state change strategy where setup change decisions are made periodically, for example, every four hours. Once a setup state of a machine is changed, it processes only a certain job type for the given time period, for example, four hours. In practice, a simulation-based method is often used to determine whether changing machine setup states or not because it can predict the future incoming WIP with a high accuracy. However, it sometimes takes a long time for a large-sized problem, which can delay the setup change decisions.

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

There have been numerous studies on determining a job sequence for such manufacturing systems with setup times. Mason et al. (2005), Pfund et al. (2008), and Topaloglu and Kilincli (2009) developed the problem-specific heuristics for scheduling complex job shops with setup times to minimize various objectives, such as total weighted tardiness and cycle time. Mönch et al. (2007) proposed a modified shifting bottleneck heuristic combined with a genetic algorithm to minimize total weighted tardiness, and Elmi et al. (2011) used a simulated annealing algorithm to minimize makespan. These heuristic methods generate an entire production schedule for a given job set, whereas dispatching rule-based approaches dynamically determine a sequence of jobs. Lee et al. (2002) compared various rule-based methods to maximize the movement, which is the number of jobs produced, of the bottleneck machines, and Chiang and Fu (2012) proposed a dispatching rule for due date-related objectives. Some studies have combined multiple dispatching rules to improve the performance of a production schedule (Dabbas et al. (2003), Lin et al. (2005), Lee et al. (2019), and Lee et al. (2020)).

These studies have considered the setups on the machines when determining a job sequence, but the setup change decision can be made separately. Sharifnia et al. (1991) and Connolly et al. (1992) proposed a dynamic policy for setup changes in single machine scheduling. Yan and Zhang (1997) considered optimal production and setup scheduling in a failure-prone manufacturing system by proposing a computational algorithm. Chung et al. (2014) proposed a framework to find a good setup change schedule for semiconductor packaging facilities. They used a genetic algorithm-based sequence optimizer, and the construction and performance evaluation of a schedule are addressed by a simulator. It can propose a proper setup change decision within a limited computation time but require a significant amount of time to evaluate several alternatives of setup state decisions with the simulation.

In this paper, we propose a machine learning-based approach by assuming a periodic setup change strategy. A neural network (NN) model, which takes the current factory state information as an input and provides a predicted value of the key performance indicator (KPI) as an output, is learned with data obtained from a simulation tool. Then a setup change decision is made by using a particle swarm optimization (PSO) method and an operation assignment rule. The NN model is used to evaluate the KPI of each solution in PSO. Our proposed method shows superior performance compared to several heuristic rules and can be applied without re-training even if the number of machines and job types are changed. We first explain the problem and the proposed approach in Sections 2 and 3, respectively. We then show the experimental results in Section 4 and provide the conclusion in Section 5.

2 PROBLEM DESCRIPTION

We consider a hybrid flow shop in which there are multiple unrelated machines in each stage. We especially focus on the bottleneck stage for the photo-lithography processes to determine whether conducting setups on the machines so that they can process other operations. There are multiple job types, each of which consists of several jobs. The jobs of the same type have the same process flow but can have different processing times and due dates. Figure 1 shows an example of a production process with reentrant flows and its Gantt chart. Figure 1(a) shows n job types in different colors that are processed in the three stages and visit stages 1 and 2 twice. In the figure, $O_{i,j,k}$ indicates the jth operation of the ith job type in the kth stage. A setup on a machine is required when not only changing the job types but also processing different operations of the same job type. For example, a setup is required between $O_{1,1,2}$ and $O_{2,1,2}$ (different job types) and also $O_{2,1,2}$ and $O_{2,2,2}$ (different operations) in stage 2 as can be seen in Figure 1(b). Hence, a setup state of a machine indicates a certain operation $O_{i,j,k}$ that the machine can process.

We assume a periodic setup change strategy in semiconductor manufacturing where setup change decisions are made periodically to reflect the practical needs of one of the semiconductor manufacturing companies in Korea. A machine in a stage has to process only a certain operation until there is a setup change. The processing sequence of the jobs in a job type on a machine is determined by a dispatcher which takes the processing times, due dates, and other features of those jobs into account.

Figure 2 shows a procedure of selecting a job with a dispatcher where $O_{i,j}$ indicates the jth operation of the ith job type and M_l is the lth machine in a stage. Since a dispatcher is operated for each stage independently, the index for a stage is omitted. The right lower matrix indicates the setup state of the five machines, which will be determined by the proposed method in Section 3. For example, M_2 and M_3 can process $O_{2,1}$, and M_1 and M_4 can handle $O_{1,2}$. After such a setup state is determined, when M_j which is set up for $O_{i,j}$ becomes idle, the dispatcher first filters out the jobs that do not require $O_{i,j}$. Then it computes the priority values for the remaining jobs and selects the one with the highest value. In the figure, when M_2, which can process $O_{2,1}$, becomes idle, jobs 2 and 4 in the buffer are filtered out, and job 1 which has a larger value than job 3 is chosen to be processed on M_2. The detailed explanation of the dispatching process can be found in Lee et al. (2018). In this paper, we focus on determining the setup states of the machines, which significantly affects the throughput of the machines.

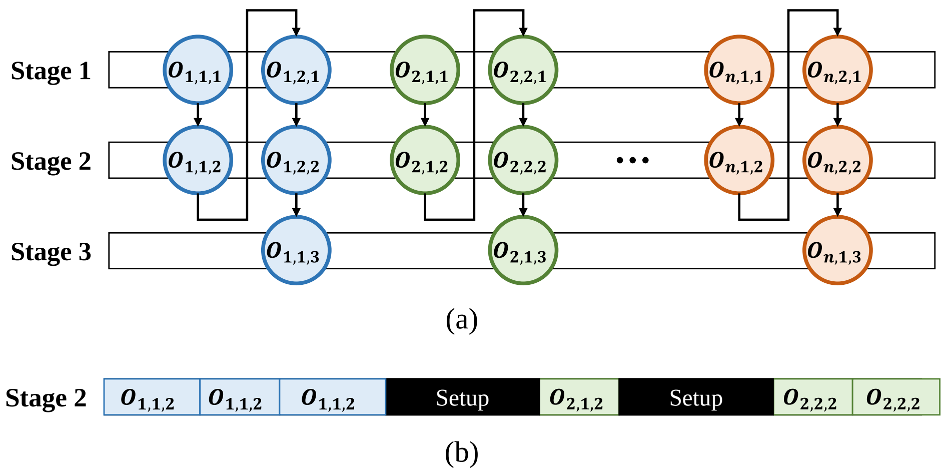

Figure 1: (a) Operation flows of n job types. (b) A production schedule for stage 2.

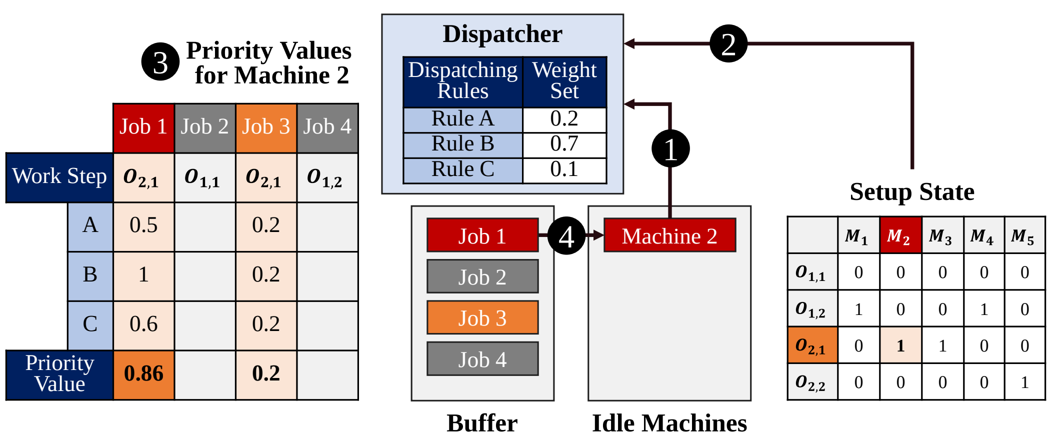

Figure 2: The procedure of a job sequence with a dispatcher.

3 SETUP CHANGE DECISION

When there are m machines and n job types, each of which visits a stage J times, the possible number of setup states in the stage is $(Jn)^m$, which requires a significant amount of time to evaluate all of the alternatives. Therefore, we provide a machine learning-based approach, for assigning certain operations to the machines, which is briefly illustrated in Figure 3. First, a prediction model is developed to estimate a KPI value, and the proportion of the number of machines assigned to each operation, defined as the machine proportion, is provided by using an optimization method, PSO in this study. The PSO improves a set of solutions iteratively and selects the best one with the highest KPI value evaluated by the prediction model. Then new setup states of the machines are derived with an assignment rule. In Figure 3, there are five machines and four operations that can be assigned. The machine proportion obtained from the prediction model-based optimizer (in Section 3.1) is given in the lower left of the figure. Then the number of machines assigned to each operation can be obtained as 0, 2, 2, and 1, respectively. After that, by considering the current setup state, the new setup state is obtained with the assignment rule in Section 3.2. This method is designed to be applicable even if the number of operations or machines is changed. The detailed explanation of the prediction model and the assignment rule is provided.

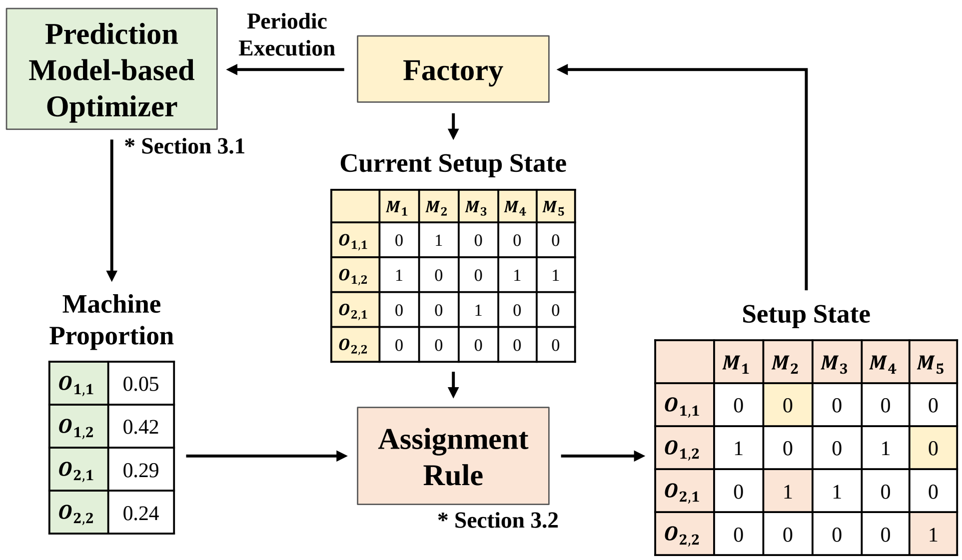

Figure 3: The proposed setup change procedure.

3.1 Prediction Model-based Optimizer

A NN model is used for predicting the KPI value and the machine proportion is obtained with PSO applied to the NN model. The following is the model we use:

$$\hat{y}(X, A) = f_\phi\left(\sum_{i=1}^{n} \sum_{j=1}^{J} g_\theta\left(x_{i,j}, a_{i,j}\right)\right),$$

where $x_{i,j}$ and $a_{i,j}$ are the inputs, for predicting the KPI value, which include the factory state information related with $O_{i,j}$ (Table 1) and the machine proportion of $O_{i,j}$ which needs to be determined, respectively. $g_\theta(\cdot)$ and $f_\phi(\cdot)$ that have the learnable parameters θ and ϕ, respectively, are the functions for transforming the input variables and providing the output value, respectively. $\hat{y}(\cdot)$ is the predicted KPI value, and X and A are the sets of $x_{i,j}$ and $a_{i,j}$, respectively. In the proposed NN model, $x_{i,j}$ and $a_{i,j}$ are concatenated and used as the input for $g_\theta(\cdot)$, and then the results of $g_\theta(\cdot)$ for each operation are aggregated into one vector, which is again used as the input for $f_\phi(\cdot)$. The structure of the NN model is shown in Figure 4. It is independent of the number of operations and the sequence of the operations due to the aggregation operation (Scarselli et al. 2008, Mao et al. 2019) and also has a smaller number of parameters to be learned (Santoro et al. 2017). The parameters in the NN model are learned with a mean squared error (MSE) loss function using the Adam optimizer (Kingma and Ba 2015), the ReLU and Softplus activation functions are used for $g_\theta(\cdot)$ and $f_\phi(\cdot)$, respectively (Glorot et al. 2011).

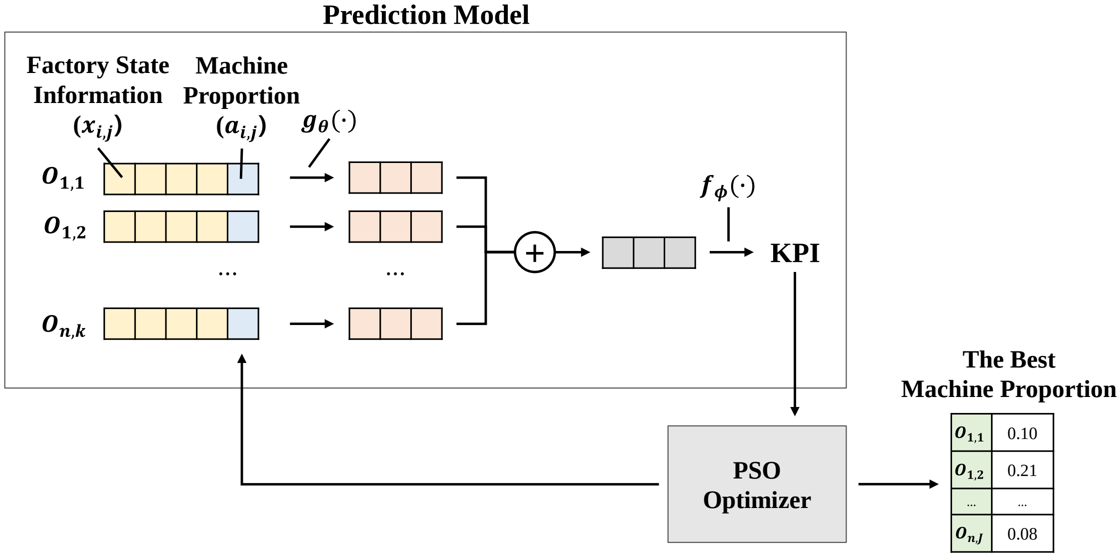

Figure 4: The architecture of the machine learning-based setup state decision method.

Table 1: Factory state information related with an operation ($x_{i,j}$).

Data	Description	Notation
Mean Processing Time	The average processing times of jobs requiring $O_{i,j}$.	$p_{i,j}$
Variance of Processing Time	The variance of the processing times of jobs requiring $O_{i,j}$.	
WIP	The number of jobs requiring $O_{i,j}$.	$w_{i,j}$
Waiting Time of WIP	The average waiting time of jobs requiring $O_{i,j}$.	
Current Machine Proportion	The current proportion of the number of machines assigned to $O_{i,j}$.	
Proportion Difference	The difference between the current and the new machine proportions of $O_{i,j}$.	
Incoming WIP	The expected number of jobs that will require $O_{i,j}$ during the next time period.	$u_{i,j}$

Table 1 shows the input variables we use for learning the NN model. They are the mean and the variance of the processing times of jobs requiring $O_{i,j}$, their current WIP and the mean waiting time before processing, the current machine proportion, the difference between the current and the new machine proportions, and the estimated incoming WIP. Note that the proportion difference and incoming WIP can be computed when $a_{i,j}$ is given, and the incoming WIP for the ith job type is calculated by considering the previous operations $O_{i,1}$,...,$O_{i,j-1}$ as in Procedure 1.

In Procedure 1, T is a period between setup change decisions, $p_{i,j}$ is the mean processing time, $q_{i,j,j-1}$ is the average of the minimum required times taken for jobs to arrive at $O_{i,j}$ from $O_{i,j-1}$, $w_{i,j}$ is the WIP of jobs requiring $O_{i,j}$, $m_{i,j}$ is the number of machines that will process $O_{i,j}$, and $u_{i,j}$ is the estimated incoming WIP for $O_{i,j}$. In the procedure, $m_{i,j}$ is computed by the assignment rule in Section 3.2 when $a_{i,j}$ is given. $u_{i,j}$ in line 6 of the procedure indicates the sum of the current WIP and the maximum number of jobs that can arrive for $O_{i,j}$ from the previous operations in the same stage. The PSO improves solutions iteratively which are evaluated with the NN model (Wu et al. 2008). Figure 4 shows this procedure.

Procedure 1: Incoming WIP calculation for the ith job type

for $j = 1, \dots, J$ **do:**
    if $j = 1$**:**
        $u_{i,j} = w_{i,j}$
    end if
    else:
        $u_{i,j} = w_{i,j} + min\left(\frac{T - q_{i,j,j-1}}{p_{i,j-1}} \times m_{i,j-1}, u_{i,j-1}\right)$
    end else
end for

3.2 Assignment Rule

We now need to assign the machines to the operations according to the machine proportion obtained in the previous subsection. Since reducing setup times can lead to improvement of movement, it is good to assign a certain operation to a machine that was already set up for the operation. We hence propose an assignment rule by considering the current setup state of each machine. The proposed rule can be applied even when the number of machines on the stage is changed. The proposed assignment rule first calculates the number of machines that need to be assigned to each operation by multiplying $a_{i,j}$ and m and then rounding it off while maintaining the sum of the number of machines assigned to all operations to m ($\sum m_{i,j} = m$).

Then it determines which machine needs to perform a setup. Let $m^c_{i,j}$ be the number of machines assigned to $O_{i,j}$ in the current setup state. In Figure 3, $m^c_{1,1}$, $m^c_{1,2}$, $m^c_{2,1}$, $m^c_{2,2}$, $m_{1,1}$, $m_{1,2}$, $m_{2,1}$, and $m_{2,2}$ are 1, 3, 1, 0, 0, 2, 2, and 1, respectively. If $m_{i,j}$ is greater than or equal to $m^b_{i,j}$, all of the $m^c_{i,j}$ machines will be assigned to the same operation again, and therefore they do not need to conduct a setup process. Otherwise, some of the $m^c_{i,j}$ machines need to change their setup states. In this case, the machine that performed the last setup at the earliest time is first chosen in turn because the quality of the jobs may deteriorate when using a machine for a long time without maintenance or setups (He et al. 2000). After determining the machines for setup changes, certain operations are assigned to the machines with the Hungarian method (Kuhn 1955) by using the assignment value of

$$\frac{T - s^l_{i,j}}{p^l_{i,j}},$$

where $s^l_{i,j}$ is the setup time occurring in M_l for $O_{i,j}$, and $p^l_{i,j}$ is the average processing time of $O_{i,j}$ in M_l. The value indicates the number of jobs that can be processed in M_l during a time period T. In Figure 3, $m^c_{1,1} - m_{1,1}$ and $m^c_{1,2} - m_{1,2}$ are both 1, and therefore M_2 and one of M_1, M_4, and M_5 should conduct a setup to process another operation. On the other hand, both $m^c_{2,1} - m_{2,1}$ and $m^c_{2,2} - m_{2,2}$ are -1, and hence $O_{2,1}$ and $O_{2,2}$ require a new machine to process them during the next period. In this case, M_2 and M_5 are assigned to $O_{2,1}$ and $O_{2,2}$ by using the Hungarian method.

4 EXPERIMENTAL RESULT

4.1 Experimental Environment

We consider a simplified semiconductor manufacturing system where there are three stages, each of which has 100, 90, and 100 machines and all job types visit the first two stages twice ($J = 2$) as illustrated in Figure 1(a). We focus on the bottleneck stage, stage 2, and all the other stages are assumed to generate schedules by using the FIFO rule. There are 10 job types ($n = 10$) in the manufacturing system, which leads to 20 possible operations. Each operation has a WIP between 0 and 3000 lots, and the average processing and setup times of the job types are 45 and 168 minutes, respectively. The KPI is the movement, which indicates the number of jobs produced in stage 2 during T of 4 hours.

The operational data for learning the prediction model is obtained with a simulation-based scheduling program for semiconductor manufacturing, MozArt, a simulation-based scheduling program used in several semiconductor fab lines in Korea (Ko et al. 2013). We generated 22,500 data for learning the prediction model by changing processing times, due dates, and WIP of job types, and setup states of machines.

We implement the NN model and the PSO optimizer in Python, and the assignment rule is implemented in MozArt with C#. We run all experiments on a PC with an Intel i9-9900 CPU @ 3.1 GHz and 64GB of RAM, and the NN model is run on via PyTorch on an NVIDIA GeForce RTX 3090 GPU. The number of nodes for a layer, number of epochs, batch size, initial learning rate of Adam optimizer, learning rate decay ratio per epoch are set to 200, 2000, 1000, 0.01, and 0.995, respectively, determined by preliminary experiments with the data generated. The number of particles, number of epochs, and inertia weight, c_1, and c_2 of the PSO optimizer are 30, 500, 0.7, 1.5, 1.5, respectively (Wu et al. (2008)).

4.2 Experimental Results

We first determine the number of layers of the prediction model. The training and validation MSEs of the NN model with a different number of layers are presented in Table 2. The performance is compared with a basic NN model, $\hat{y}(X,A) = f_{\phi'}(X,A)$. We use 5-fold cross-validation and present the average MSE. In Table 2, it can be observed that the proposed model has a smaller MSE than the basic model, and it has the smallest MSE with two layers. We note that the mean absolute percentage error (MAPE) of the validation and test sets is about 5 %.

Table 2: MSE of the prediction model with a different number of layers.

Model	Number of Layers	Training MSE	Validation MSE
Basic model	1	7196.5	8822.9
	2	7420.5	8588.1
Proposed model	1	7946.9	8612.5
	2	4394.5	**5810.2**
	3	4392.1	6156.1

We now compare the proposed method with two heuristic rules; (Current Setup) one is to keep the current machine setup states, and (WIP Ratio) the other is to determine the number of machines for each operation by considering the current WIP and to assign them with the proposed assignment rule. The proposed approach considers not only the current WIP and incoming WIP but also the current machine proportion and average processing times of job types.

Table 3: Movement (and setup times) for different WIP levels.

Average Level of WIPs	Current Setup	WIP Ratio	Proposed Approach
2	**8300.8** (52.6)	4920.1 (124.5)	6490.1 (98.1)
1.5	**7696.8** (31.6)	4854.8 (123.1)	7388.8 (65.8)
1	5104.5 (16.3)	4710.9 (128.5)	**8232.4** (23.0)
0.8	5058.3 (9.0)	4758.1 (128.4)	**7156.2** (36.7)
0.5	3564.1 (1.7)	4482.9 (140.4)	**6340.4** (31.8)
0.3	2796.3 (0)	4696.7 (143.0)	**6280.4** (24.9)
0.1	1950.8 (0)	4188.0 (158.0)	**4440.4** (42.3)
0.05	800.3 (0)	**1834.4** (156.3)	1736.3 (110.3)
0.02	172.6 (0)	**1388.9** (135.3)	1214.3 (117.9)
Average	3938.3 (12.4)	3981.6 (137.5)	**5475.5** (61.2)

Table 3 shows the movement from the two heuristics and the proposed approach regarding the different levels of WIPs. Note that the NN model was learned with the data according to the average WIP level of 1. It is observed that the movement of the proposed approach is larger than the two heuristic rules for the instances with WIP levels less than 1. This is because the first heuristic, Current Setup, causes some idle machines with a lower number of WIPs, and the WIP Ratio requires many setups on the machines.

We then compare the results of the methods when the number of machines or job types is changed. Each of the results is shown in Table 4 and Table 5, respectively. When the number of machines is changed as in Table 4, the effect of the assignment rule can be observed because the proportion of machines assigned does not change. It is observed that the proposed method performs better than the other two heuristics in Tables 4 and 5.

Table 4: Movement (and setup times) for number of machines in stage 2.

Number of Machines in Stage 2	Current Setup	WIP Ratio	Proposed Approach
100	5245.0 (25.4)	5154.2 (145.9)	**8666.1** (28.1)
95	5216.8 (19.1)	4904.3 (138.2)	**8366.6** (28.5)
85	4822.6 (14.3)	4376.8 (122.0)	**7722.5** (22.8)
80	4726.9 (12.5)	4090.1 (113.5)	**7221.0** (24.1)
Average	5002.8 (17.8)	4631.4 (129.9)	**7994.0** (25.9)

Table 5: Movement (and setup times) for different number of job types.

Number of Job Types (n)	Current Setup	WIP Ratio	Proposed Approach
12	5162.7 (28.3)	4858.5 (119.7)	**8824.8** (10.4)
8	5719.0 (14.8)	4644.5 (133.0)	**6581.0** (70.2)
Average	5440.8 (21.6)	4751.5 (126.4)	**7702.9** (40.3)

5 CONCLUSION

We have developed the machine learning-based approach that determines the setup states of the machines by assuming a periodic setup decision strategy. The NN model that considers various factory states as inputs and provides the KPI value as output was learned with the operational data generated from a simulation tool. Then a PSO optimizer using the NN model for evaluating the fitness values of solutions searches the best machine assignment proportion, and the proposed assignment rule assigns machines to jobs. The experimental results have shown that the proposed approach performs better than two heuristic methods and can still provide a proper setup decision even if the number of machines and job types are changed. More extensive experiments for complex manufacturing systems are required with the proposed method. In addition, an optimization method instead of PSO can be used to find the optimal machine setup assignment.

ACKNOWLEDGMENTS

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2019R1C1C1004667).

REFERENCES

Chiang, T.-C. and Fu, L.-C. 2012. "Rule-based Scheduling in Wafer Fabrication with Due Date-based Objectives". C*omputers & Operations Research* 39(11):2820-2835.

Chung, B. S.; Lim, J.; Park, I. B.; Park, J.; Seo, M.; and Seo, J. 2014. "Setup Change Scheduling for Semiconductor Packaging Facilities using a Genetic Algorithm with an Operator Recommender". *IEEE Transactions on Semiconductor Manufacturing* 27(3):377-387.

Connolly, S.; Dallery, Y.; and Gershwin, S. B. 1992. "A Real-time Policy for Performing Setup Changes in a Manufacturing System". In *Proceedings of the 31st IEEE Conference on Decision and Control*. December 16th-18th, Tucson, Az, USA.

Dabbas, R. M.; Fowler, J. W.; Rollier, D. A.; and Mccarville, D. 2003. "Multiple Response Optimization using Mixture-designed Experiments and Desirability Functions in Semiconductor Scheduling". *International Journal of Production Research* 41(5):939-961.

Elmi, A.; Solimanpur, M.; Topaloglu, S.; and Elmi, A. 2011. "A Simulated Annealing Algorithm for the Job Shop Cell Scheduling Problem with Intercellular Moves and Reentrant Parts". *Computers & Industrial Engineering* 61(1):171-178.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. "Deep sparse rectifier networks". In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics.* April 11th-13th, Lauderdale, FL, USA.

He, S.; Nishiki, H.; Hartzell, J.; and Nakata, Y. 2000. "Low Temperature PECVD a-Si:H TFT for Plastic Substrates". In *SID Symposium Digest of Technical Papers* 31(1):278–281. Oxford: Blackwell Publishing Ltd.

Kingma, D. P. and Ba, J. 2015. "Adam: A Method for Stochastic Optimization". In *Proceedings of the International Conference on Learning Representations (ICLR).* May 7th-9th, San Diego, CA, USA.

Ko, K.; Kim, B. H.; and Yoo, S. K. 2013. "Simulation based Planning and Scheduling System: MozArt®". In *Proceedings of the 2013 Winter Simulation Conference*, edited by Pasupathy, R.; Kim, S.-H.; Tolk, A.; Hill, R.; and Kuhl, M. E., editor(s), pages 4103-4104. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kuhn, H. W. 1955. "The Hungarian Method for the Assignment Problem". *Naval Research Logistics Quarterly* 2(1-2):83-97.

Lee, J.-H.; Kim, Y.; Kim, Y.-B.; Kim, B.-H.; Chung, G.-H.; and Kim, H.-J. 2020. "A Sequential Search Method of Dispatching Rules for Scheduling of LCD Manufacturing Systems". *IEEE Transactions on Semiconductor Manufacturing* 33(4):496-503.

Lee, J.-H.; Kim, Y.; Kim, Y.-B.; Kim, H.-J.; Kim, B.-H.; and Chung, G.-H. 2018. "A Framework for Performance Analysis of Dispatching Rules in Manufacturing Systems". In *Proceedings of the 2018 Winter Simulation Conference,* edited by Rabe, M.; Juan, A. A.; Mustafee, N.; Skoogh, A.; Jain, S.; and Johansson, B., editor(s), pages 3550-3560. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lee, J.-H.; Kim, Y.; Kim, Y.-B.; Kim, H.-J.; Kim, B.-H.; and Chung, G.-H. 2019. "A Sequential Search Framework for Selecting Weights of Dispatching Rules in Manufacturing Systems". In *Proceedings of the 2019 Winter Simulation Conference,* edited by Mustafee, N.; Bae, K.-H.G.; Lazarova-Molnar, S.; Rabe, M.; Szabo, C.; Haas, P.; and Son, Y.-J., editor(s), pages 2201-2211. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lee, Y. H.; Park, J. W.; and Kim, S. Y. 2002. "Experimental Study on Input and Bottleneck Scheduling for a Semiconductor Fabrication Line". *IIE Transactions* 34:179-190.

Lin, J. T.; Wang, F. K.; and Kuo, P. C. 2005. "A Parameterized-dispatching Rule for a Logic IC Sort in a Wafer Fabrication". Production Planning & Control 16(5):426-436.

Mao, H.; Schwarzkopf, M.; Venkatakrishnan, S. B.; Meng, Z.; and Alizadeh, M. 2019. "Learning scheduling algorithms for data processing clusters". In *Proceedings of the ACM Special Interest Group on Data Communication.* August 19th-23th, Beijing, China.

Mason, S. J.; Fowler, J. W.; Carlyle, W. M.; and Montgomery, D. C. 2005. "Heuristics for Minimizing Total Weighted Tardiness in Complex Job Shops," *International Journal of Production Research* 43(10):1943-1963.

Mönch, L.; Schabacker, R.; Pabst, D.; and Fowler, J. W. 2007. "Genetic Algorithm-based Subproblem Solution Procedures for a Modified Shifting Bottleneck Heuristic for Complex Job Shops". *European Journal of Operational Research* 177(3):2100-2118.

Pfund, M. E.; Balasubramanian, H.; Fowler, J. W.; Mason, S. J.; and Rose, O. 2008. "A Multi-criteria Approach for Scheduling Semiconductor Wafer Fabrication Facilities". *Journal of Scheduling* 11(1):29-47.

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. "A Simple Neural Network Module for Relational Reasoning". In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, December 4th-9th, Long Beach, CA, USA.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. "The graph neural network model". *IEEE transactions on neural networks* 20(1):61-80.

Sharifnia, A.; Caramanis, M.; and Gershwin, S. B. 1991. "Dynamic Setup Scheduling and Flow Control in Manufacturing Systems". *Discrete Event Dynamic Systems* 1(2):149-175.

Topaloglu, S. and Kilincli, G. 2009. "A Modified Shifting Bottleneck Heuristic for the Reentrant Job Shop Scheduling Problem with Makespan Minimization". *The International Journal of Advanced Manufacturing Technology* 44(7):781-794.

Wu, J.; Zhang, C. H.; and Cui, N. X. 2008. "PSO Algorithm-based Parameter Optimization for HEV Powertrain and its Control Strategy". *International Journal of Automotive Technology* 9(1):53-59.

Yan, H. and Zhang, Q. 1997. "A Numerical Method in Optimal Production and Setup Scheduling of Stochastic Manufacturing Systems". *IEEE Transactions on Automatic Control* 42(10):1452-1455.

AUTHOR BIOGRAPHIES

JE-HUN LEE is a Ph.D. student in Department of Industrial & Systems Engineering, Korea Advanced Institute of Science and Technology (KAIST). He received a MS in industrial engineering from Sungkyunkwan University. He is interested in scheduling methodologies and applications and operations management. His email address is jehun.lee@kaist.ac.kr.

HYUN-JUNG KIM is an Assistant Professor with the Department of Industrial & Systems Engineering, Korea Advanced Institute of Science and Technology (KAIST). She received B.S., M.S., and Ph.D. in industrial and systems engineering from KAIST. Her research interests include discrete event systems modeling, scheduling, and control. Her email address is hyunjungkim@kaist.ac.kr.

YOUNG KIM is a Ph.D. student in Department of Industrial Engineering, Sungkyunkwan University. He is interested in simulation-based modeling and scheduling. His email address is lmjlguard@gmail.com.

YUN BAE KIM is a Professor with the Department of Industrial Engineering, Sungkyunkwan University. He received the MS degree from the University of Florida, and the Ph.D. degree from Rensselaer Polytechnic Institute. His current research interests are demand forecasting, simulation methodology, high tech market analysis and scheduling. His email address is kimyb@skku.edu.

BYUNG-HEE KIM is the President of VMS Solutions Co., Ltd.. He received a BS from Sungkyunkwan University, MS and Ph.D. from Korea Advanced Institute of Science and Technology (KAIST) all in industrial engineering. He is interested in simulation-based scheduling and planning, manufacturing information systems, BPMS, and virtual manufacturing. His email address is kbhee@vms-solutions.com.

GU-HWAN CHUNG is a Head Researcher of VMS Solutions Co., Ltd.. He received a MS in industrial engineering from Korea Advanced Institute of Science and Technology (KAIST). He is interested in simulation-based scheduling and planning, manufacturing information systems, BPMS, and virtual manufacturing. His email address is chunggh@vms-solutions.com.