Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

A SEQUENTIAL SEARCH FRAMEWORK FOR SELECTING WEIGHTS OF DISPATCHING RULES IN MANUFACTURING SYSTEMS

Je-Hun Lee
Young Kim
Yun Bae Kim
Hyun-Jung Kim

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

Byung-Hee Kim
Gu-Hwan Chung

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

ABSTRACT

Dynamic manufacturing systems consisting of multiple stages use a combination of dispatching rules to obtain production schedules in general. A weighted sum method, which assigns different weights to each dispatching rule and prioritizes jobs with a high weighted sum, is widely used especially for LCD and semiconductor manufacturing. A suitable set of weights by considering dynamic system states has to be determined in order to improve the throughput and utilization of systems. Hence, we develop a sequential search framework, with simulation and decision trees, which can generate a good weight set of dispatching rules within a short period of time. We show that the proposed search method performs better than a random search by performing experiments with real fab data.

1 INTRODUCTION

A typical LCD fab line consists of multiple process stages, each of which has parallel machines and a buffer, and where jobs are processed in the stages sequentially and visit some stages multiple times. The scheduling problem of the fab line corresponds to a flexible job shop problem (FJSP) which is NP-hard (Garey et al. 1976). The manufacturing environment for LCD is very dynamic because of various types of products, reentrant flows, and material handling between stages. Hence, such production lines are mostly operated with dispatching rules to determine a sequence of jobs.

The performance of a schedule obtained with a single dispatching rule such as SPT (Shortest Processing Time) or EDD (Earliest Due Date) is low in general because of dynamics of manufacturing environments and various production requirements, and hence, fab engineers have developed special rules to represent their knowledge about operations and used a different set of dispatching rules for each stage. A set of dispatching rules used in each stage is determined by reflecting the characteristics of the stage such as large processing times or setup times, and small capacity of buffers. A priority-based method which sorts jobs in a buffer based on a certain standard has been used (Zhang and Rose 2013 and Lee et al. 2018) in some manufacturing environments, whereas a weighted sum method, which assigns different weights to each rule and prioritizes jobs with a high weighted sum, has been widely used in many semiconductor fab lines in Korea. The weighted sum method is more useful than the priority-based one because a variety of weights can provide diverse solutions. However, a suitable set of weights depending on system states has to be determined and adjusted periodically in order to improve the throughput and utilization of the systems.

A typical LCD fab line is operated with three shifts a day, and engineers adjust weights for dispatching rules every shift regularly by considering the current production environments. Since engineers fully rely on their experiences for setting weights of dispatching rules, the performance, such as throughput or setup times, varies depending on the shifts. Hence, it is required to propose a good set of weights to engineers by considering current system's states such as machine states, demand, processing times of jobs, setup times and due dates of jobs. Since production environments keep changing, historical data obtained from different conditions of manufacturing systems are not useful to analyze the relationship between weights and performances. Hence, it is necessary to find a set of weights, within a short period of time, by simulating the fab line reflecting real-time states of processing stages.

There have been many studies on searching such weights or hyper-parameters that can provide optimal objective values. A grid search is one of traditional search methods that explores all combinations of variables in a finite search space, but the computation time increases exponentially if there are many variables to be searched. A random search refers to searching parameters for a limited time or with a small number of samples when a search space is large or unlimited. Bergstra and Bengio (2012) showed that the random search is more effective than the grid search in estimating parameters in artificial neural networks (Caflisch et al. 1997).

Latin hypercube sampling (LHS) is a statistical method for generating random samples of parameters distributed uniformly (McKay et al. 1979). The method divides a search space into $n \times n$ sub-spaces of the same size, and then selects n sub-spaces as samples. Park (1994) proposed optimal Latin hypercube sampling (OLHS) which chooses sub-spaces so that they are uniformly and symmetrically distributed in an entire search space.

Bernardo et al. (1992) and Welch and Sacks (1991) proposed a sequential design strategy for searching parameters. In the strategy, an approximate function of a target variable is derived from given data, and then the function is updated by adding sampled data points sequentially. Jones et al. (1998), Schonlau et al. (1998) and Lehman (2002) used the Bayesian model and Gaussian process to obtain a function that describes the relationship between input variables and a dependent variable (or a target variable). The Bayesian model generates several functions that describe the given data, and the sequential design strategy uses the average of the functions to estimate the value of the dependent variable. In each iteration, a data point that can improve the accuracy of the model the most is chosen, and then the several functions are updated with the new data point. The disadvantage of the method is a large computation time for deriving the function in a high-dimensional search area.

Simpson et al. (2001) and Huang et al. (2006) proposed a metamodel-based optimization method using kriging to search for the global optimal solution. The proposed method differs from the sequential design in criteria for sampling new data. The sequential design increases the accuracy of the approximation model iteratively, whereas the metamodel-based optimization method focuses on the regions with a high probability of having the global optimal solution based on the estimated kriging model. In order to construct the metamodel, Wang (2005) and Yang et al. (2007) used the neural network, and Wang and Simpson (2004) used the response surface method and kriging method sequentially.

Both methods, the sequential design strategy and metamodel-based optimization method, have the high performance in exploring efficient solutions in a large region, but are difficult to apply to practical scheduling environments because they require a large computation time.

We hence propose a search framework using a decision tree that can provide a good set of weights for dispatching rules quickly. The framework adopts a sequential design strategy and minibatch data selection to effectively reduce the search space by eliminating areas with a low probability of having optimal values. In the rest of this paper, we describe the scheduling problem using dispatching rules in Section 2, and propose a sequential search framework in Section 3. Then we show the performance of the framework in Section 4.

2 PROBLEM DESCRIPTION

We consider an LCD fab line in which schedules are generated using the weighted sum of multiple dispatching rules. There is a virtual model of this line, implemented with the MozArt, which allows fab engineers to obtain production schedules. MozArt is an integrated development and operations solution that can implement production planning and scheduling applications with a virtual model created by abstraction of real manufacturing factories (Ko et al. 2013). The MozArt dispatcher process based on the weighted sum of multiple dispatching rules is as follows:

1. When a machine becomes idle, call the dispatcher.
2. The dispatcher assigns jobs certain values from each dispatching rule.
3. A weighted sum of each job is computed with a given weight of dispatching rules.
4. The job with the highest weighted sum is assigned to the machine. Ties are broken with the first-in-first-out (FIFO) rule.

Suppose that jobs 1 and 2 are waiting in a buffer of a process (or a machine) where dispatching rules A and B are used. Assume that job 1 obtains the values of 0.4 and 0.2 from dispatching rules A and B, respectively, whereas job 2 receives 0.1 and 0.8 from the rules, and the weights of 0.7 and 0.3 for the two dispatching rules are given. Then the weighted sums of jobs 1 and 2 are 0.34 (=0.7×0.4+0.3×0.2) and 0.31 (=0.7×0.1+0.3×0.8), respectively, and job 1 is assigned to the machine.

The LCD fab line consists of TFT, CF, and Cell shops, and each shop is composed of multiple stages that have identical machines in each stage. We consider the TFT shop among them. The performance of a photo-lithography process is most important because it is one of the bottleneck processes in the TFT shop. However, the performance of the TFT shop is not only determined by the schedules of the photo-lithography process but also by other processes. One of these processes is a deposition process, which is the second most significant process in the efficiency of the TFT shop, located at a step just before the photo-lithography process. Therefore, we determine the weights of dispatching rules of both photo-lithography and deposition processes and apply the FIFO rule to other processes. The buffer capacity of each process is not limited.

The dispatching rules used in the LCD fab line have been developed by fab engineers to reflect the characteristics of the dynamic environments of LCD manufacturing (Lee et al. 2018). The dispatching rules considered in the photo-lithography and deposition processes are shown in Table 1. In the photo-lithography process, min move quantity (MMQ) rule, prevent frequent setup (PFS) rule, proportion lot type (PLT) rule, FIFO rule, and target delay (TD) rule are used. In the deposition process, the same rules are applied except for the PFS rule. Fab engineers set the weights of dispatching rules based on their knowledge and experience, which may not yield optimal results. The framework we propose can suggest good weights of dispatching rules to obtain schedules with the high performance, which can lead to efficiency improvement of the factory.

We consider the setup times and throughput of the machines of the photo-lithography process as the key performance indicators (KPIs) of the schedule. The consecutive operations of jobs of the same type can improve the throughput of the photo-lithography process and overall processes.

Table 1: Dispatching rules for the photo-lithography and deposition processes (Lee et al. 2018).

Dispatching Rule	**Description**	**Score Type**	**Process**
Min Move Quantity	Assign 1 to a job if the same job type is being processed on a machine.	binary	photo-lithography deposition
Prevent Frequent Setup	Assign 1 to a job if the number of units that stay in the buffer is larger than or equal to a given target.	binary	photo-lithography
FIFO	Assign a large value to a job that arrives earlier than others.	continuous	photo-lithography deposition
Proportion Lot Type	Assign a large value to a job type that has a large number of units in the buffer.	continuous	photo-lithography deposition
Target Delay	Assign a large value to a job if it is urgent.	discrete	photo-lithography deposition

3 SEQUENTIAL SEARCH FRAMEWORK FOR A WEIGHT SET

When the KPIs the user considers are set, the proposed sequential search framework can suggest the weights of dispatching rules to derive high KPI values. Since there is no sufficient time for determining weight sets in practice, our framework divides the entire search space into sub-spaces and focuses on certain sub-spaces where high KPI values can be found. The search space is divided by a decision tree every iteration. The process of the proposed search framework is as follows (refer to Figure 1):

Step 1: Sample initial weight sets with a sampling method (Section 3.1).
Step 2: Collect KPIs for the schedules generated with the sampled weight sets by simulation (Section 3.2).
Step 3: Learn a decision tree from weight sets and KPIs and then segment the search space of weights based on the decision tree (Section 3.3).
Step 4: Assign a certain number of samples for each segmented search space based on KPI values in each segmented sub-space (Section 3.4).
Step 5: Generate weight sets using a probabilistic selection method for each segmented space (Section 3.5).
Step 6: Repeat Steps 2-5 until a termination condition is met.

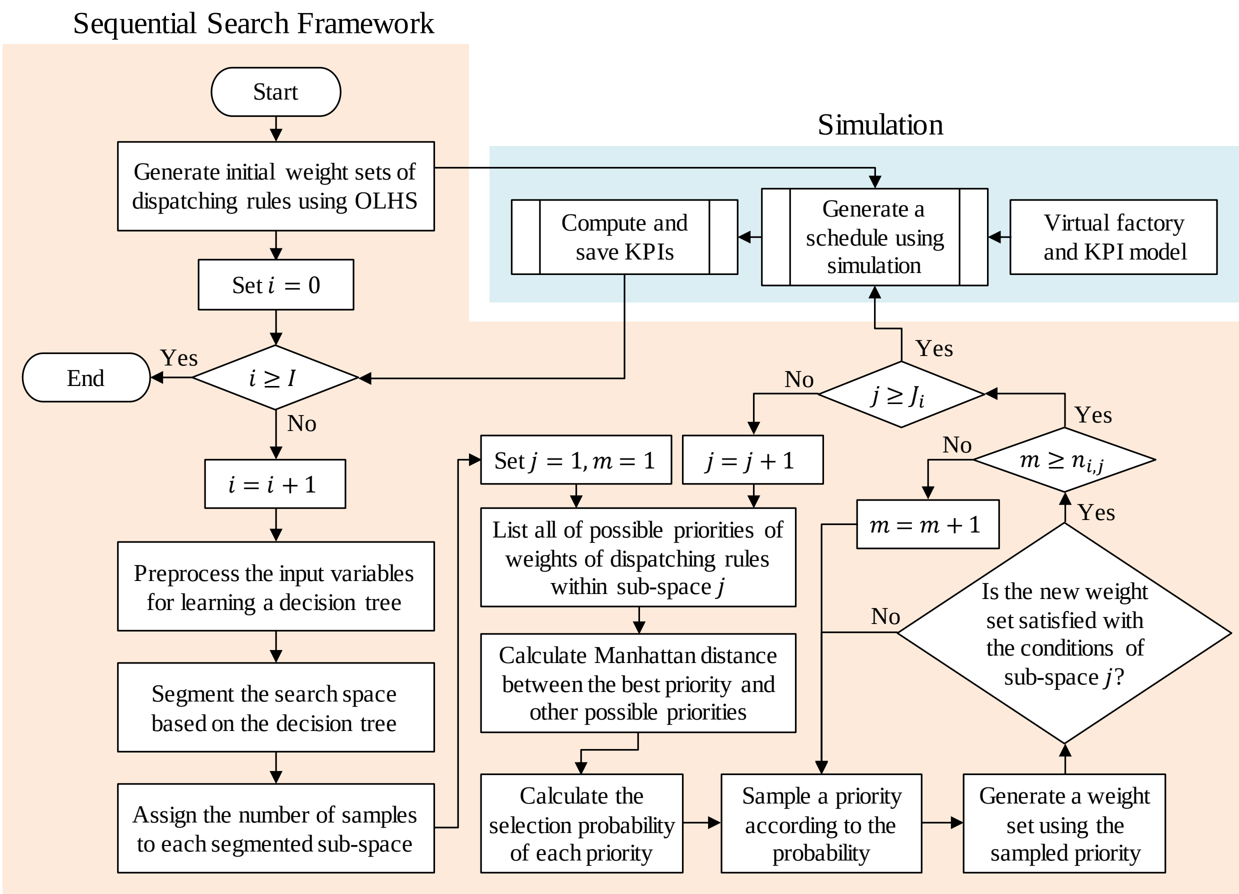

Figure 1: Flow chart of sequential search framework for a weight set of dispatching rules.

After the weight search framework is completed, the best weight set of dispatching rules based on the KPI values is provided to engineers. The input parameters of the framework, such as the initial number of samples, n_0, the number of iterations, I, and the number of samples per iteration i, n_i, where $1 \leq i \leq I$, are set in advance. We now describe each step of the framework in detail.

3.1 Sampling for an Initial Weight Set

In the initial sampling step, the entire space should be searched because there is no information on the relationship between KPIs and parameters (or weights). However, due to a time limit, a large initial sample size cannot be used, and hence we use the OLHS method that selects parameters evenly (Park 1994). Then, n_0 weights for the five dispatching rules of the photo-lithography process and n_0 weights for the four dispatching rules of the deposition process are sampled with OLHS.

3.2 Schedule Generation with MozArt-based Simulation

In this step, KPIs for each schedule derived from the weights generated in Step 1 or Step 5 are computed by MozArt, the simulation-based scheduling program. Before the simulation runs, the weights generated in each step are rescaled to be used in the simulation model. The range of weights is set to [1, 10^5] reflecting real fab environments.

3.3 Search Space Segmentation with a Decision Tree

By learning a decision tree with all KPI data, we can obtain several sub-spaces bounded with some criterion. Decision tree learning is a non-parametric supervised learning method that divides the data set into subsets according to simple decision rules inferred from the data and predicts the values of a target variable (or a KPI) for new input data. The constructed model of the decision tree is relatively easy to understand compared to other machine learning techniques because the format of conditions of a subset is boolean logic and human-readable, and it expresses the interaction of various input variables. Thus, a decision tree can be used to visually and explicitly represent sub-spaces of an entire search space. We use a decision tree learning method proposed by Breiman et al. (1984) to classify the search area of the weight sets. In the proposed method, a regression model that learns a continuous target value generates a branch based on the variation reduction.

We use the weights of each dispatching rule and their combination as input variables of the decision tree. Since the weight ratio in each stage is correlated with the scheduling performance, the actual input value is converted into the ratio of the weights of the same stage, so that the sum of the values corresponding to each stage is 1. Table 2 shows input variables used in learning a decision tree (Dabbas et al. 2003). In the table, x_A and x_B indicate weights of a pair of dispatching rules.

Table 2: Input variables of a decision tree.

Category	Input variable	No. of variables used for the photo-lithography process	No. of variables used for the deposition process
Weight of each dispatching rule	x_A	5	4
Ratio of two weights in the same process	$x_A \,/\, x_B$	10	6
Multiplication of two weights in the same process	$x_A \times x_B$	10	6
Exponential of weight	$\exp(-x_A)$	5	4

3.4 Number of Samples Assigned to a Sub-space

The framework decides the number of samples selected in each sub-space segmented in Step 3 to generate n_i weight sets for iteration i. For all J_i sub-spaces separated from the decision tree, the performance measure P_j is calculated for each sub-space j, where $1 \le j \le J_i$, based on the KPI values of the corresponding space. P_j is computed by the weighted sum of the mean and the best KPI obtained in each sub-space j, as shown in equation (1):

$$P_j = \alpha \bar{y}_j + \beta y_j^*, \tag{1}$$

where $\bar{y}_j$ and y_j^* are the mean of the KPI and the best KPI in sub-space j, respectively, and α and β are weights for $\bar{y}_j$ and y_j^*, respectively. The KPI values are normalized with the min-max method, and the sum of α and β is 1, which are determined experimentally.

After calculating P_j for a total of J_i sub-spaces, the ratio K_j of weight sets in each space is determined with the normalization shown in equations (2) and (3):

$$R_j = \left[\{\max(P_j) - P_j\}/\{\max(P_j) - \min(P_j)\}\right]^{\gamma} \tag{2}$$

$$K_j = R_j / \textstyle\sum_{k \in J_i} R_k, \tag{3}$$

where R_j is a weight assigned to each sub-space j and depends on the value of γ. As γ increases, the difference among R_j values of sub-spaces becomes larger, which leads to assigning a higher weight to the space with large KPIs. γ is also determined experimentally. The number of weight sets of sub-space j in iteration i ($n_{i,j}$) is determined by the ratio K_j, and $\sum_{j \in J_i} n_{i,j}$ is equal to n_i.

3.5 Weight Set Generation with Segmented Conditions of Sub-space

In the last step, $n_{i,j}$ weight sets assigned to each sub-space j are probabilistically generated within the segment conditions. Lee et al. (2018) showed that the priority of dispatching rules in each process significantly influences the performance of a schedule through extensive experiments. Hence, weight sets are generated by following the priority of dispatching rules that provides the best KPI value, which was selected in the Step 4. The process of generating weight sets is as follows:

- Step 5.1: List the priorities of weights of dispatching rules for each process that can be searched in each sub-space j.
- Step 5.2: For each sub-space j, calculate the weighted Manhattan distance between the priority of the weight set that yielded the best KPI and the priorities listed in Step 5.1.
- Step 5.3: Assign the score for each priority based on the calculated weighted Manhattan distance and then transform it to a sampling probability value.
- Step 5.4: Sample a priority according to the calculated probability.
- Step 5.5: Generate a weight set that satisfies the selected priority and the segment conditions of the sub-space.
- Step 5.6: Repeat Steps 5.4-5.5 until generating $n_{i,j}$ weight sets assigned to each sub-space j.

Suppose that weight set 1 has the best KPI value in a certain sub-space where dispatching rules A, B and C are used. Assume that rule B has the highest weight and rule C has the smallest one in the set. Then the priority of set 1 can be denoted as (2 ,1, 3). Then, the weighted Manhattan distance between the best priority and all of possible priorities can be computed with the weight of 2, 3, and 1 for rules A, B, and C, respectively. When the best priority of three rules is (2 ,1, 3), the example of the weighted Manhattan distance and sampling probability is described in Table 3. In Step 5.3, the values are transformed into probabilities by scaling them with a min-max normalization method and then by computing a ratio of the scaled value to the sum of all values. The priority of dispatching rules which has the largest weighed distance is given with the sampling probability of 0. After that, a weight set is generated by reflecting the probabilistically selected priority.

This process repeats every iteration and the framework can search more improved weight sets as the number of iterations increases. After repeating steps 2-5 of the framework until the number of iterations I set in advance is reached, the weight set of the best performing schedule is suggested to the fab engineer.

Table 3: An example of computing weighted Manhattan distance with three rules.

Best priority: (2, 1, 3)		
List of all possible priorities of weights	Weighted Manhattan distance	Sampling probability
(1, 2, 3)	1 × 2 + 1 × 3 + 0 × 1 = 5	0.18
(1, 3, 2)	1 × 2 + 2 × 3 + 1 × 1 = 9	0
(2, 1, 3)	0 × 2 + 0 × 3 + 0 × 1 = 0	0.41
(2, 3, 1)	0 × 2 + 2 × 3 + 2 × 1 = 8	0.05
(3, 1, 2)	1 × 2 + 0 × 3 + 1 × 1 = 3	0.27
(3, 2, 1)	1 × 2 + 1 × 3 + 2 × 1 = 7	0.09

4 PERFORMANCE OF SEARCH FRAMEWORK

We consider the setup times and throughput of photo-lithography machines as the KPIs of schedules and search the weight sets of photo-lithography and deposition processes using the proposed framework and also the random search method for the comparison. MozArt generates a schedule of three days with a set of given weights and computes KPIs.

In order to evaluate the performance of the proposed framework, we generate three virtual factory models (Cases 1, 2, 3) using the facilities and operations data of the actual LCD fab line in China. The three virtual models follow the same production environment that has 20 parallel machines of the photo-lithography process and 16 parallel machines of the deposition process. All of the three models have seven product types but their production quantities are 770, 839, and 906, respectively, which corresponds to three days of production, and the standard deviations for each production quantity are 466, 422, and 583, respectively.

The time constraint is set to 3 hours which allow 45 weight sets to simulate factory operations using MozArt on a PC with an Intel® Core™ i5-4460 3.2 GHz processor with 5Gb RAM. In the random search, the weight of each dispatching rule is randomly chosen between 0 and 10^5 to be used in the simulation model. The hyper-parameters of the sequential search framework was determined by considering the time limit and preliminary experiments. The parameters, n_0, n_i and I, were set to 5, 8, and 5, respectively, and α, β and γ were set to 0.5, 0.5, and 5, respectively.

The experimental results obtained from ten runs of each case is presented in Tables 4 and 5 for the setup times and throughput, respectively. The proposed framework performed better than the random search in terms of the average and standard deviation of KPI values. In Table 4, the average sum of setup times of three days of production with the proposed framework is 1911.74 minutes whereas the random search generates 1958.98 minutes. The framework reduced the sum of setup times by 2.41% compared to the random search. The small deviation of the KPI values of the proposed framework means that the performance is robust.

Table 5 compares the throughput of photo-lithography machines. The mean throughput of three days of production with the proposed framework is 373,082.7 lots whereas it is 370,617.3 lots with the random search. The throughput is increased by 0.67% with the proposed framework. The percentile rank of KPI values obtained from the proposed framework and random search among 450 randomly generated samples is denoted in Table 6. The proposed framework provides setup times and throughput at the 99.57th percentile on average among 450 samples whereas the random search provides the KPI values at the 98.54th percentile on average.

Table 4: The sum of setup times of photo-lithography machines with the 3 hour limit (min).

Exp. No.	Case 1		Case 2		Case 3	
	Random	**Framework**	**Random**	**Framework**	**Random**	**Framework**
1	2,220.5	2,255.0	1,909.4	1,866.2	1,641.6	1,555.2
2	2,341.4	2,160.0	1,935.4	1,935.4	1,684.8	1,710.7
3	2,073.6	2,142.7	2,116.8	1,900.8	1,762.6	1,676.2
4	2,255.0	2,246.4	2,013.1	1,849.0	1,779.8	1,658.9
5	2,324.2	2,246.4	1,788.5	1,969.9	1,581.1	1,615.7
6	2,289.6	2,168.6	1,788.5	1,969.9	1,745.3	1,598.4
7	2,246.4	2,272.3	1,892.2	1,900.8	1,667.5	1,702.1
8	2,220.5	2,116.8	1,926.7	1,900.8	1,831.7	1,650.2
9	2,332.8	2,160.0	2,047.7	1,866.2	1,658.9	1,529.3
10	2,237.8	2,194.6	1,900.8	1,840.3	1,555.2	1,693.4
Average	2254.2	**2196.3**	1931.9	**1899.9**	1690.8	**1639.0**
Standard deviation	78.3	**54.6**	104.6	**46.5**	88.4	**62.6**

Table 5: The throughput of photo-lithography machines with the 3 hour limit (ea).

Exp. No.	Case 1		Case 2		Case 3	
	Random	**Framework**	**Random**	**Framework**	**Random**	**Framework**
1	403,780	408,400	393,933	391,873	314,553	317,720
2	406,467	408,307	395,907	398,980	311,487	316,907
3	401,720	409,367	393,360	397,893	315,993	313,767
4	403,753	400,793	397,207	400,427	313,233	319,880
5	403,333	408,287	399,233	390,333	313,867	314,247
6	400,040	404,807	390,320	402,313	311,327	318,813
7	406,340	408,280	391,653	392,193	313,467	314,300
8	411,620	407,493	391,673	399,073	316,533	319,587
9	406,547	409,367	389,767	397,073	312,273	314,527
10	402,787	404,933	391,780	393,207	314,567	309,333
Average	404,639	**407,003**	393,483	**396,336**	313,730	**315,908**
Standard deviation	3,248	**2,708**	**3,098**	4,121	**1,752**	3,285

Table 6: Percentile rank of the weight set among the 450 random data (%).

KPI	**Method**	**Case 1**	**Case 2**	**Case 3**	**Average**
Setup times	Search framework	**99.7**	**99.4**	**99.6**	**99.57**
	Random search	98.5	98.6	98.1	98.40
Throughput	Search framework	99.4	99.6	99.7	**99.57**
	Random search	98.7	98.5	98.8	98.67

If the time limit is set to 8 hours, our computation environment allows 120 weight sets to simulation. The parameters, n_0, n_i and I, are set to 50, 10, and 7, respectively, and α, β and γ are the same as before. Table 7 shows the result of the throughput of photo-lithography machines with the 8 hour limit. The mean throughput of three days of production with 120 runs (8 hour limit) is 373,842.6 lots, which is increased by

0.20% compared to the result of 45 runs (3 hour limit). The throughput is increased by 0.45% with the proposed framework compared to the performance of the random search with the 8 hour limit.

Table 7: The throughput of photo-lithography machines with the 8 hour limit (ea).

Exp. No.	**Case 1**		**Case 2**		**Case 3**	
	Random	**Framework**	**Random**	**Framework**	**Random**	**Framework**
1	406,467	404,926	395,907	393,234	314,553	317,910
2	403,753	408,257	397,207	396,806	315,993	319,857
3	406,340	408,893	399,233	401,307	313,867	314,750
4	411,620	406,413	391,673	394,713	316,533	316,660
5	402,787	406,393	391,780	400,960	314,567	316,560
Average	406,193	**406,976**	395,160	**397,404**	315,103	**317,147**
Standard deviation	3,431	**1,595**	**3,351**	3,635	**1,113**	1,888

5 CONCLUSION

We developed a sequential search framework that can provide good weight sets in a small amount of time. We considered the setup times and throughput of photo-lithography machines as the KPIs, and determined the weight sets of photo-lithography and deposition processes. With the 3 hour limit, the proposed framework provides a weight set that improved the setup times by 2.41% and the throughput by 0.67% on average compared to the random search method. In addition, the throughput is improved by 0.20% on average when the time constraint is relaxed to 8 hours. The decision tree method effectively segmented the search space, and the framework was able to focus on the segments with a high probability of having good weights. More experiments with different time constraints, sophisticated rules, and various parameters should be performed to verify the performance of the framework.

The framework is expected to improve the operational performance of various manufacturing systems using dispatching rules and can be used to search for higher dimensional parameters. Also, this framework can be applied equally if several KPIs are considered and their relative importance is determined. However, in many cases, it is difficult to quantify the relative importance of KPIs, and hence multi-objectives should be considered.

ACKNOWLEDGMENTS

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2019R1C1C1004667), by the Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korean government (MOTIE) (No. 20003957), by the ICT R&D program of IITP/MSIT [B0101-17-1081], development of ICT based software platform and service technologies for medical 3D printing applications, and by a National IT Industry Promotion Agency (NIPA) grant funded by the Korean government (MSIP)(S0249-19-1004), prediction based smart SCM framework for offshore industry.

REFERENCES

Bergstra, J. and Y. Bengio. 2012. “Random Search for Hyper-parameter Optimization”. *Journal of Machine Learning Research* 13(1): 281-305.

Bernardo, M. C., R. Buck, L. Liu, W. A. Nazaret, J. Sacks, and W. J. Welch. 1992. “Integrated Circuit Design Optimization using a Sequential Strategy”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 11(3):361-372.

Breiman, L., J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and Regression Trees*. 1st ed. London: Chapman and Hall/CRC Press, Inc.

Caflisch, R. E., W. J. Morokoff, and A. B. Owen. 1997. "Valuation of Mortgage Backed Securities using Brownian Bridges to Reduce Effective Dimension". *Journal of Computational Finance* 1:27-46.

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

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. "The Complexity of Flowshop and Jobshop Scheduling". *Mathematics of Operations Research* 1:117–129.

Huang, D., T. T. Allen, W. Notz, and N. Zeng. 2006. "Global Optimization of Stochastic Black-box Systems via Sequential Kriging Meta-models". *Journal of Global Optimization* 34: 441-466.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black–box Functions". *Journal of Global Optimization* 13(4): 455–492.

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

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

Lehman, J. S. 2002. *Sequential Design of Computer Experiments for Robust Parameter Design*. Ph.D. thesis, Department of Statistics, The Ohio State University, Columbus, Ohio.

McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. "Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". *Technometrics* 21(2):239-245.

Park, J. S. 1994. "Optimal Latin-Hypercube Designs for Computer Experiments". *Journal of Statistical Planning and Inference* 39(1): 95-111.

Schonlau, M., W. J. Welch, and D. R. Jones. 1998. "Global versus Local Search in Constrained Optimization of Computer Models". *Lecture Notes-Monograph Series* 34(1):11–25.

Simpson, T. W., T. M. Mauery, J. J. Korte, and F. Mistree. 2001. "Kriging Models for Global Approximation in Simulation-based Multidisciplinary Design Optimization". *American Institute of Aeronautics and Astronautics*, 39(12):2233-2241.

Wang, G. G. and T. Simpson. 2004. "Fuzzy Clustering based Hierarchical Metamodeling for Design Space Reduction and Optimization". *Engineering Optimization* 3(3):313-335.

Wang, L. 2005. "A Hybrid Genetic Algorithm–Neural Network Strategy for Simulation Optimization". *Applied Mathematics and Computation* 170(2):1329-1343.

Welch, W. J. and J. Sacks. 1991. "A System for Quality Improvement via Computer Experiments". *Communications in Statistics-Theory and Methods* 20(2):477-495.

Yang, F., B. Ankenman, and B. L. Nelson. 2007. "Efficient Generation of Cycle Time-Throughput Curves Through Simulation and Metamodeling". *Naval Research Logistics* 54(1):78-93.

Zhang, T. and O. Rose. 2013. "Intelligent Dispatching in Dynamic Stochastic Job Shops". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 2622-2632. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

JE-HUN LEE is a Ph.D. student in Department of Industrial Engineering, Sungkyunkwan University. He is interested in scheduling theory, algorithms, and applications in manufacturing systems and material handling systems. He is also interested in applying reinforcement learning methods to scheduling problems. His email address is swi02050@gmail.com.

YOUNG KIM is a Ph.D. student in Department of Industrial Engineering, Sungkyunkwan University. He is interested in simulation-based modeling and scheduling, simulation-based analysis methods, and quantum computing. 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 quantum computing. His email address is kimyb@skku.edu.

HYUN-JUNG KIM was an Assistant Professor with the Deptartment of Industrial Engineering, Sungkyunkwan University. She holds a Ph.D. in industrial and systems engineering from Korea Advanced Institute of Science and Technology (KAIST). She is now an Assistant Professor with the Department of Industrial & Systems Engineering, KAIST. Her research interests include discrete event systems modeling, scheduling, and control, especially for manufacturing systems. Her email address is

kim.hj@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 all in industrial engineering. He is interested in simulation-based scheduling and planning, manufacturing information systems, business process management systems, virtual manufacturing, and autonomous response systems. His email address is kbhee@vms-solutions.com.

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