IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

An Imperialist Competitive Algorithm Incorporating Remaining Cycle Time Prediction for Photolithography Machines Scheduling

PENG ZHANG [1], XINMING ZHAO [2], XIA SHENG [1], AND JIE ZHANG [3]

1 Institute of Intelligent Manufacturing and Information Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

2. Institute of Design and Control Engineering for Heavy Equipment, Shanghai Jiao Tong University, Shanghai 200240, China

3 College of Mechanical Engineering, Donghua University, Shanghai 201620, China

Corresponding author: Jie Zhang (mezhangjie@dhu.edu.cn).

This work was funded by the National Natural Science Foundation of China under Grant No. U1537110.

ABSTRACT Photolithography machines are the common bottleneck in semiconductor manufacturing system. The operation constraints in photolithography machines are very complicated, including wafers arriving over time, dedicated machine constraints for critical layers, auxiliary resources constraints and dynamic manufacturing environment. In previous researches, the dynamic manufacturing environment has never been considered, which would make remaining cycle time seriously deviate from the expected value, and then result in the deterioration of scheduling performance. In this paper, an imperialist competitive algorithm incorporating remaining cycle prediction is proposed for photolithography machines scheduling problem with the objective of total completion time minimization. A deep autoencoder neural network is presented at first to predict remaining cycle time, responding to the environmental changes. Secondly, an imperialist competitive algorithm in the framework of rolling horizon strategy is proposed to address the scheduling problem, incorporated with the accurately predicted remaining cycle time. Several procedures are designed to improve the performance of the algorithm. To verify the proposed algorithm, a simulation model of semiconductor manufacturing system is constructed and numerical tests are conducted in the model. Results show that the algorithm proposed can significantly decrease wafers' average cycle time.

INDEX TERMS photolithography machines; imperialist competitive algorithm; remaining cycle time prediction; semiconductor manufacturing

I. INTRODUCTION

The semiconductor industry is a highly expensive and complex manufacturing system that requires an enormous capital investment. Because of economic consideration, a short cycle time is pursued to achieve a fast recovery of this capital. The equipment investment of photolithography machines is the highest in semiconductor manufacturing system, and as such act as a bottleneck in the system. It is observed that about one-third of the total WIP (Work-In-Process) competes at the photolithography machines in a large-scale memory fabrication line [1], which leads the machines to be the major factor to affect the system's performance. Hence, photolithography machines scheduling has been the main concern for production control managers.

In photolithography area, wafers are placed onto the photolithography machines and a specific pattern is placed over them. The flow of wafers through photolithography machines is re-entrant, meaning that wafers visit the machines several times to print several layers of patterns in wafers. Typically, mix-and-match strategies are adopted to reduce the high cost of photolithography machines, to use leading-edge (and expensive) machines only for the critical layers, while using cheaper tools for other layers. Then the processing times and production ranges are different for different machines. Since each photolithography machine has its own calibration characteristics, which will impact the alignment of patterns between different critical layers. All the critical layers of a wafer are required to be processed on the same machine, i.e. dedicated machine constraints. Besides, every photo process requires the availability of a specific mask, which serves as an auxiliary resource to print a circuit pattern, and is both product- and layer-dependent. Only when

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

the mask is in presence can the wafer be processed. After completing one photo process, wafers will continue to go through other processes such as etch and implantation, and then revisit the photolithography machines again. However, a great many dynamic events, such as machines breakdown and urgent orders inserting, may happen when wafers leave photolithography area, which will make wafers' remaining cycle time seriously deviate from the expected value. All of these constraints, namely wafers arriving over time, dedicated machine constraints for critical layers, auxiliary resources constraints and dynamic manufacturing environment, makes photolithography machines scheduling an extremely complicated task.

The photolithography machines scheduling problem has aroused much attention and been studied by academic and industrial researchers. Cakici and Mason [2] developed two different heuristic solution approaches for photolithography machines scheduling, taking mask requirements as the auxiliary resource constraints. Shr et al. [3] proposed the heuristic Load Balancing (LB) scheduling approach based on a Resource Schedule and Execution Matrix (RSEM) to tackle the dedicated machine constraints. Zhou et al. [4] used Kohonen neural network to select best combination dispatching rules for dynamic scheduling of photolithography process. Ham and Cho [5] presented an integration of real-time dispatching with linear programming technique considering the transportation and auxiliary resource constraints. It is worth noting that these studies mainly focus on real-time dispatching rules for photolithography machines scheduling with specific constraints, which cannot achieve optimal solutions or adapt to the dynamic environment. Therefore, artificial intelligence optimization methods known as meta-heuristic algorithms began to attract researchers' interest. Zhou [6] proposed an effective estimation of distribution algorithm for photolithography machines scheduling problems with auxiliary resource constraints. Bitar et al. [7] exploited memetic algorithm to solve the scheduling problem with auxiliary resources in a photolithography workshop of a semiconductor plant. However, they both used the fixed time horizon, which cannot adapt to the dynamic production environment. Zhang et al. [8] developed a variable time interval based rolling horizon strategy to address the dynamic arriving wafers and proposed an imperialist competitive algorithm (ICA) to solve the scheduling problem. They drew the conclusion that ICA is the best by comparing the total completion time of several algorithms, including dispatching rules and meta-heuristic algorithms. Nevertheless, they haven't considered the dynamic manufacturing environment.

Leonardi and Raz [9] showed that remaining cycle time is the main factor in the design of algorithm to minimize total flow time and cycle time. However, the dynamic manufacturing environment, including machines breakdown and urgent inserting orders in other processes, will greatly affects wafers' remaining cycle time and make it uncertain. Therefore, we should accurately predict remaining cycle time and incorporate it into the scheduling algorithm. Researches related to the problem mainly focus on the prediction of wafers' cycle time and the scheduling algorithm incorporating prediction algorithm. Ankenman et al. [10] predicted wafers' cycle time via simulation on demand. They can just obtain the mean steady-state cycle time, while every wafer's flow time is different. Meidan et al. [11] applied conditional mutual information maximization for feature selection and the selective naïve Bayesian classifier for cycle time prediction. Wang and Zhang [12] proposed a conditional mutual information-based feature selection process to select key feature subset and presented a concurrent back-propagation network based forecasting model to predict the cycle time of wafers. Chen and Wang [13] proposed a nonlinearly normalized back propagation network to predict wafers' cycle time. However, these researches selected the key features mainly by experience or informatics formulas, which may leads to information loss and repeated calculation in the dynamic environment. Besides, a classification algorithm should be designed at first and different categories are then learned with different networks. Hung et al [14] experimented with two flow time prediction methods—an exponential smoothing method and an empirical queueing approach, and examined two dispatching rules—the modified least slack rule and the shortest remaining flow time rule to reduce flow times for wafer fabrication facilities. Chen and Wang [15] proposed a nonlinear scheduling rule incorporating a fuzzy-neural remaining cycle time estimator to improve scheduling performance in a semiconductor manufacturing factory. Li and Yu [16] built a remaining cycle time prediction model by random forest algorithm and designed a scheduling strategy of semiconductor production lines with remaining cycle time prediction. However, the scheduling algorithm in the research above are belonging to dispatching rules, which are myopic. Antoencoders are a type of deep neural network that can be used to reduce data dimensionality and have been shown to yield state-of-the-art performance in a variety of tasks ranging from object recognition and learning invariant representations to syntactic modeling of text [17]. Some of the highly successful examples of non-probabilistic feature learning models are autoencoder networks. Therefore, in this paper, a deep autoencoder network is employed for remaining cycle time prediction. Metaheuristic algorithms deal with complicated scheduling problems by providing near optimal solutions within the reasonable amount of time [18]. Imperialist competitive algorithm (ICA) is a meta-heuristic algorithm based on imperialistic competition where the populations are represented by countries and has shown great optimization ability in various scheduling problems [19-23].

This paper proposes an improved imperialist competitive algorithm incorporating remaining cycle time prediction for photolithography machines scheduling. The rest of the paper

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

is organized as follows. Section 2 gives the definition of the problem. The proposed algorithm can be found in section 3. Section 4 is the computational results and discussion. Finally, conclusions are outlined in Section 5.

II. PROBLEM DEFINITION

A. PROBLEM DISCRIPTION

Photolithography machines scheduling problem can be represented by $\mathrm{Rm}\,|\,\mathrm{aux},\ \mathrm{ded},\ \mathrm{rcrc},\ \mathrm{online}|\sum C_j$ employing the notation of Pinedo [24], where Rm means unrelated parallel machines, aux indicates auxiliary resource constraints, ded is dedicated machine constraints, rcrc means that wafers visit photolithography machines more than once, online represents that wafers arrive over time and each wafer is unknown until its arrival time. Then the problem can be stated as follows. A set of wafers have to be processed in lots, with a size of 25 wafers, in a single production stage composed by many unrelated parallel machines, aiming at minimizing the total completion time (TCT). Each lot has a different processing time and an uncertain remaining cycle time, and belongs to a kind of product. Each kind of product has its own photo times, and only one mask is available for each photo process. We need to assign different lots to different machines and sequence them with unknown remaining cycle time. The lot can be processed only when a photolithography machine and the required photomask are both available. After that, the wafer is put into other areas for other processes, and the lot will return to the photolithography area for the next layer processing and so on.

B. Mathematical Formulation

1) NOTATIONS

Parameters

J The set of lots to be processed, $J=\{1,2,\ldots,j,\ldots\}$.

L The set of processing layers, $L=\{1,2,\ldots,l,\ldots\}$.

δ_l The set of lots requiring processing layer l.

K The set of photolithography machines, $K=\{1,2,\ldots,k,\ldots\}$

K_j The set of photolithography machines that lot j can be processed.

p_{jk} The processing time of lot j on machine k, $k \in K_j$.

r_j The arrival time of lot j.

v_j The dedicated machine of lot j, $v_j=0$ if process layer of lot j is the first critical layer.

f_j The remaining cycle time of lot j after current photo process.

M A big number.

Decision variables

C_j Completion time of lot j.

x_{ijk} 1 if lot i immediately precedes lot j on machine k; otherwise, it is 0, $i, j \in J$, $k \in K$.

e_{ij} 1 if lot i finishes its processing before lot j starts its processing, $\forall i, j \in \delta_l$, $\forall l \in L$, $i \neq j$.

2) MODEL FORMULATION

The mixed integer programming model for the problem is below.

$$\min\ TCT = \sum (c_j + f_j) \tag{1}$$

Constraints:

$$\sum_{j\in J:\, j\neq 0} \sum_{k\in K} x_{0jk} \le |K| \qquad \forall i \in J, i \neq 0 \tag{2}$$

$$\sum_{j\in J:\, j\neq 0} \sum_{k\in K} x_{j0k} \le |K| \qquad \forall i \in J, i \neq 0 \tag{3}$$

$$\sum_{j\in J:\, j\neq i} \sum_{k\in K} x_{jik} = 1 \qquad \forall i \in J, i \neq 0 \tag{4}$$

$$\sum_{j\in J:\, j\neq i} \sum_{k\in K} x_{ijk} = 1 \qquad \forall i \in J, i \neq 0 \tag{5}$$

$$\sum_{j\in J:\, j\neq i} \sum_{k\in K, k\notin K_i} x_{jik} = 0 \qquad \forall i \in J, i \neq 0 \tag{6}$$

$$\sum_{j\in J:\, j\neq i} x_{jiv_i} = 1 \qquad \forall i \in J, i \neq 0, v_i \neq 0 \tag{7}$$

$$\begin{aligned} &C_i - C_j + x_{ijk}(M + p_{ijk}) \le M \\ &\forall i, j \in J, \forall k \in K, j \neq 0, i \neq j \end{aligned} \tag{8}$$

$$\begin{aligned} &C_i \ge r_i + p_{ik} + \sum_{j\in J, i\neq j} (\max(0, r_j + p_{jk} - r_i)) x_{jik} \\ &\forall i \in J, i \neq 0, \forall k \in K \end{aligned} \tag{9}$$

$$e_{ij} + e_{ji} \ge 1 \forall i, j \in \delta_l, \forall l \in L, j \neq 0, i \neq j \tag{10}$$

$$x_{ijk} \in \{0,1\} \tag{11}$$

$$e_{ij} \in \{0,1\} \tag{12}$$

The objective (1) minimizes the total completion time. Constraints (2) and (3) assume that each machine starting and ending its schedule with a dummy lot 0. Constraints (4) and (5) dictate machine assignment and job sequencing for lots on the same machine. Constraints (6) ensures that each lot j can only be processed on the machine in K_j. Constraint (7) ensures that all the critical layers of a lot should be processed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

on one machine. Constraint (8) guarantees that one machine should process a lot immediately after another. Constraint (9) ensures that a lot can be processed only when the lot is released and the required machine is available. Finally, constraint (10) ensures that only one of the lots with the same processing layer can be processed a time, considering the photomask number constraints. Constraints (11) and (12) define the range of the variables.

III. IMPROVED ICA ALGORITHM WITH REMAINING CYCLE TIME PREDICTION

An improved ICA algorithm incorporating remaining cycle time prediction is proposed for photolithography machines scheduling, as can be seen in FIGURE 1. At first, a deep autoencoder neural network is developed to predict the remaining cycle time of lots in different photolithography machines schedules. Then a rolling horizon strategy rolling horizon strategy is used to decide the scheduling point and the lots to be scheduled. Finally, an improved ICA algorithm is presented to schedule the selected lots with the predicted schedule-dependent remaining cycle time.

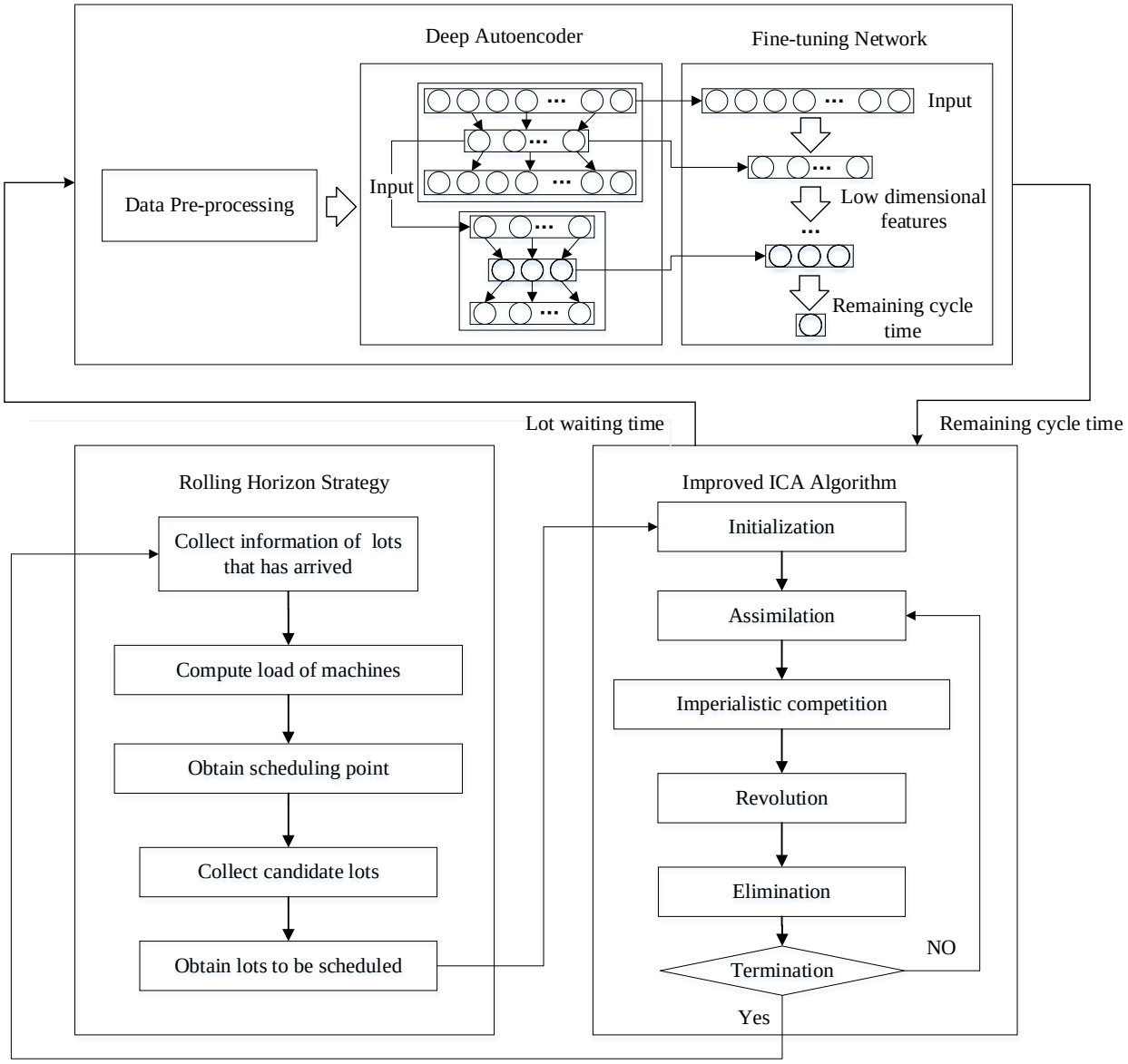


FIGURE 1. **Framework of the ICA algorithm Incorporating Remaining Cycle Time Prediction**

A. Deep Autoencoder Neural Network based Prediction Model

1) DATA PRE-PROCESSING

Wang et al. [12] split the candidate features for cycle time prediction into two classes, order related features and workshop related features. However, we should consider not only the current information but also the future information for remaining cycle time prediction because wafers revisit the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

photolithography machines several times and current processing priority will impact the all subsequent processing. Therefore, the candidate features in this paper are described in TABLE 1.

TABLE 1. Description of each candidate feature

Type		Variables	Description
Current status	Order related	$TP_1,...,TP_n$	The processing times for process1,…, process n of lots
		$CTP_1,...,CTP_n$	The processing times completed for process1,…, process n of lots
		$\mathrm{Pr}_1,...,\mathrm{Pr}_n$	The priority of lots
	Workshop related	$Q_1,...,Q_n$	The waiting queue length of machines
		$U_1,...,U_n$	The utilization of machines
		N_{wip}	The total number of work in process
Future status	Scheduling related	$WT_1,...,WT_n$	The waiting times for lot 1, ,lot n in photo area in the schedule

After extracting the data of candidate features from the manufacturing execution systems (MES), data normalization should be made to make sure that error reduction rate in the back propagation of neural networks keeps consistent along the direction of each dimension. Data normalization can be made by (2):

$$x_i^{Normalized} = (x_i - x_{\min}) / (x_{\max} - x_{\min}) \quad (13)$$

Where x_i is current sample value, $x_{\min}$ and $x_{\max}$ are the minimum and maximum value of sample.

2) PRE-TRAINING BY DEEP AUTOENCODER NEURAL NETWORK

A deep neural network is utilized to construct the complex relationship between the features and wafers' remaining cycle time. As the structure of the model grows deeper, however, the neural network trained by back-propagation would be prone to falling into the local minima, leading to a rather poor prediction accuracy. This problem can be alleviated by choosing a comparatively better initial value of the weight and the bias, which could be achieved by layer-wise pre-training. According to the conclusion of [25], it is more effective to train several simple models beforehand and concatenate them greedily. Although the greedy policy does not necessarily (and most likely not) ensure a global optimum solution, its solution would be more competitive. As a result, using an autoencoder to optimize the training procedure of a deep neural network is a feasible solution to prevent the network from falling into the local minima.

Autoencoder, which is a three-layer neural network, is composed of one input layer, one hidden layer and one output layer. The input and the output of autoencoder are identical, hence, the main aim of autoencoder is to learn the parameters of the network by minimizing the reconstruction error between the input and its duplicated output. The mathematical form of its target can be summarized as:

$$\theta, \theta' = \arg\min_{\theta,\theta'} \left(\frac{1}{2}\left\| h_{\theta'}(f_\theta(X)) - X \right\|^2\right) \quad (14)$$

Where $f_\theta(X)$ denotes the non-linear map from input feature space towards hidden layer feature space, which is the activation function value of the hidden units. The mapping function is the Sigmoid function. More specifically, $f_\theta(X)$ is defined by:

$$f_\theta(X) = \frac{1}{1+e^{-(W^T X + b)}}, \theta \in \{W, b\} \quad (15)$$

Where W is the weight matrix of the edge connecting the input units and the hidden units, and b is bias vector, both of which are the main parameters determined by back-propagation process. The final output which derives from the activation function value of the hidden units is defined in the same way.

In the most of the autoencoder structure, the number of the hidden units is smaller than the number of the input units, forcing the autoencoder to learn a condensed low-dimensional representation of the input feature space. However, it is ineffective and impossible to learn the most refined representation by a single autoencoder under the circumstance where the input dimension is high. For high-dimensional problem, a deep autoencoder must be constructed greedily. The structure of a deep autoencoder is shown in FIGURE 1.

As is illustrated in the figure, a single hidden layer autoencoder is trained to learn a lower-dimension representation of the input feature space for the first time. After that, a second-time dimensionality reduction process is achieved by using the activation function value of the hidden units from the first autoencoder as the input value of the second autoencoder. By stacking and training several autoencoders layer by layer, a much more condensed representation of the input would be learned by deep autoencoders with a relatively lower reconstruction error. So far, the pre-training process has completed.

3) FINE-TUNING NETWORK

After the pre-training process, the weight matrixes between each two layers of the deep autoencoder are extracted from the final iteration, which are utilized to initial the weight matrixes in the fine-tuning network for predicting wafers' remaining cycle time. It can be inferred that the output of the final hidden layer of the prediction network in the first iteration of the feed forward process would be the low-dimensional representation learned by deep autoencoders, which preserves the information from the input and simultaneously makes the learning process easier because of the its smaller dimensionality. Apart from the pre-training process, elaborately choosing the hyperparameters of the fine-tuning network and using some generalization tricks can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

also be a great help to improve the generalization capability in the test set.

Network structure parameters. The number of hidden units, especially the one of the last hidden units, has a great effect on the final performance, which should be determined firstly throughout the empirical formula:

$$l = \sqrt{m+n} + a \tag{16}$$

Where l determines the number of the last hidden units, m denotes the number of the input units, and n denotes the number of the output units. a is the adjustable value ranging from -5 to 5 commonly.

Optimization algorithms. Empirically, the learning algorithm with adaptive learning rates such as 'Adams' performs more robustly and converges faster under non-convex scenarios [25]. The learning algorithm chosen in this paper is the 'Adams'.

Generalization-relative parameters. To prevent overfitting problem, a penalty parameter is introduced into the loss function, which limits the value of weights and bias to achieve simple but generalized model. The revised loss function is defined as:

$$L = \frac{1}{2}\left\| h_{\theta'}(f_{\theta}(X)) - X \right\|^2) + \lambda(\left\|\theta\right\|^p + \left\|\theta'\right\|^p) \tag{17}$$

Where λ is the generalization parameter, which introduces sparsity into the weight matrixes when p is valued as 1 or restricts the energy of weight matrixes when p is as 2. The optimal number of λ could be around 0.01 or 0.001, empirically.

Convergence parameters. Learning rate is a classic convergence-related parameter, which is commonly set to be an exponentially decayed value by iteration.

B. Rolling Horizon Strategy

In a semiconductor manufacturing factory, tens of thousands of wafers are processed simultaneously in the system. However, in most semiconductor manufacturing factory, roughly 35-45% of wafers working in process are resident in the photolithography area [26]. Then the number of scheduling jobs in photolithography machines will be huge. In this research, a rolling horizon strategy is adopted to decompose the photolithography machines scheduling problem into several local scheduling problems along the time horizon. Each local scheduling problem can get wafers' local processing information, including the next process, candidate processing machines, processing time and dedicated machines. At each decision point, scheduling job set should be selected and local schedule should be made.

Here, we define a machine load threshold T_0 , and scheduling will be triggered at the point when the cumulative processing time of arrival lots is a multiple of T_0 or there is an idle machine.

After that, we use a threshold $N_{\max}$ to limit the scheduling size. When the scheduling task is triggered, we count the remaining total number of lots N in the candidate scheduling job set, and select $\min\{N, N_{\max}\}$ lots according to their urgency, and add them to the scheduling job set.

C. Improved ICA Algorithm

The ICA algorithm is a novel metaheuristic algorithm to solve optimization problems. Atashpaz [27] proposed this algorithm for continuous optimization problem. Since then, several researches showed great performance from ICA for combinatorial optimization problems. ICA algorithm starts with an initial population, called countries which can be divided into two categories: imperialists and colonies. Then one imperialist, together with several colonies, can form an empire. And imperialistic competition will happen among different empires, during which weak empires would collapse and powerful empires can take possession of their colonies. Finally, there will exists only one empire. The motivation behind the algorithm is the imperialist competition among countries in sociopolitical evolutionary behavior of countries. The implementation of original ICA mainly includes five steps.

Step 1 (Initialization): Generate a number of countries, select the powerful ones to be the imperialists, assign the remaining countries to the imperialists on the basis of their power and establish the initial empires.

Step 2 (Assimilation): Colonies move toward their corresponding imperialist. Exchange the position of colony and imperialist in an empire when the colony is more powerful than imperialist.

Step 3 (Imperialist competition): Empires compete for the weakest colony from the weakest empire. An empire will be removed if it loses all colonies.

Step 4 (Revolution): Replace several weak colonies with new ones randomly.

Step 5 (Termination): If stopping criteria meet, stop, if not go back to step 2.

As mentioned before, after the scheduling lots are selected, an improved ICA algorithm is proposed to handle the scheduling task.

1) ENCODING AND DECODING PROCEDURES

Encoding. An individual consists of two segments is designed for the unrelated parallel machines scheduling problem, in which the first segment dictates the machine assignment and the second one determines the lot sequencing. A random number between 0 and 1 is used to encode the first segment $x = [x_1, x_2, \ldots, x_n]$, $0.0 < x_j < 1.0$, where n is the lot number to be scheduled. Then $r_j = \lceil x_j * | K_j | \rceil$ is the index of machine selected for lot j . The second segment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

is $y=[y_1,y_2,\ldots,y_n], y_j \in \{1,2,\ldots,n\}$. Then if $y_k = j$, lot j is the k th lot to be processed.

Decoding. We can obtain the completion time of all lots by decoding the two segments as follows:

For $j=1,\ldots,n$, do

Step 1) Selected y_i from the second segment, where $y_i=j$, and Compute the machine r_i for lot i with x_i ;

Step 2) If the process for lot i is critical layer but not the first, set $r_i = v_i$.

Step 3) If photomask required is available for lot i when machine r_i become idle at t , then computer its completion time $C_i = t + p_{ir_i}$; otherwise, select y_k from the second segment, where $y_k=j+1$, and set $y_i=j+1$ and $y_k=j$. Go to step 1);

Step 4) Refresh the load σ_{r_i} allocated to machine r_i as follows.

$$\sigma_{r_i}=\begin{cases}\sigma_{r_i}+\varepsilon_i-1, \text{if the process is the first critical layer}\\ \sigma_{r_i}-1, \text{if the process is critical layer but not the first}\\ \sigma_{r_i}, \text{otherwise}\end{cases} \tag{18}$$

Where, ε_i is the critical layer number of lot i .

2) INITIALIZATION PROCEDURE

ICA starts with a population of N_{pop} countries, each of which represents a feasible solution for the scheduling problem. With the encoding procedure mentioned before, we can randomly generate the initial population.

For $p=1,\ldots,N_{pop}$, do

Step 1) Generate the first segment.

For $j=1,\ldots,n$

If the process for lot j is critical layer but not the first, set $x_j = v_j / |K_j|$. Otherwise, divide the interval $[0,1]$ into K_j intervals, i.e. $[0,1/|K_j|)$, $[1/|K_j|, 2/|K_j|)$,..., $[(|K_j|-1)/|K_j|, 1)$. Generate a random number $x_j \in [(k-1)/|K_j|, k/|K_j|)$ by possibility $(\sigma^{\max}_{[j]}-\sigma_{[k]})/\sum_{i=1}^{|K_j|}(\sigma^{\max}_{[j]}-\sigma_{[i]})$, where $\sigma^{\max}_{[j]}$ is the maximum load of the candidate machines for lot j , $\sigma_{[k]}$ is the load of the k th candidate machine for lot j .

Step 2) Generate the second segment

Randomly generate an array $y=[y_1,y_2,\ldots,y_n], y_j \in \{1,2,\ldots,n\}$, each y_j appears only once.

According to the procedure above, the first segment ensures that machines with heavier load would be selected with less probability.

After generating the initial population of size N_{pop} , we can obtain the completion times and waiting times of all lots according to the decoding procedure for every country. Moreover, we can predict the remaining cycle time of all lots according to current information of system and the scheduling result. Finally, we can compute the cost of every country as follows:

$$OF_p = \sum_{j=1}^{n}(C_j + f_j) \tag{19}$$

Then selecting N_{imp} countries with the lowest cost as imperialists. To form the N_{imp} empires, the colonies are divided among the imperialists based on their power. The number of colonies in p th empire can be obtained by:

$$Ncol_p = round\{V_p \cdot N_{col}\} \tag{20}$$

Where V_p is the normalized power of p th imperialist and N_{col} is the number of colonies. V_p is defined by:

$$V_p = \left| \frac{NOF_p}{\sum_{i=1}^{N_{imp}} NOF_i} \right| \tag{21}$$

Where NOF_p is the normalized power of p th imperialist which can be achieved as follows:

$$NOF_p = \max\{OF_i\} - OF_p \tag{22}$$

3) ASSIMILATION POLICY

Assimilation is an operator taking a colony toward its imperialist in which colony can inherit from both the colony and imperialist. We adopt an adaptive assimilation policy to weaken the learning effect in the early stage while strengthen the learning effect in the late state, trading off population diversity and convergence rate [28]. At first, we generate a binary vector with the density of the ones equal to β . And then the genes of the new colony corresponding to one will copy the genes of the imperialist. Parameter β varies as follows:

$$\beta = \beta_{\max} - \frac{(MaxDecad - Decade)\times(\beta_{\max}-\beta_{\min})}{MaxDecad - 1} \tag{23}$$

Where $\beta_{\min}$ and $\beta_{\max}$ are the minimum and maximum value of β , $MaxDecad$ and $Decade$ represent the maximum iteration number and current iteration number.

4) COMPETITION MECHANISM

In imperialistic competition process, empires more powerful will conquer colonies of weaker empires. The power of an empire is composed of the cost of imperialist and colonies.

$$TOF_p = OF_p + \zeta \cdot mean\{OF_i\} \tag{24}$$

Where ζ is a coefficient between 0 and 1 to reduce the effect of colonies cost.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

According to socio-political theory, the probability of each empire conquering the weakest colony of the weakest empire bases on not only its power but also its distance to the colony. Similarly, the normalized cost of the empire is:

$$NTOF_p = \max\{TOF_i\}\text{-}TOF_p \quad (25)$$

And the normalized distance is given by:

$$NDis_p = \max\{Dis_i\}\text{-}Dis_p \quad (26)$$

Where Dis_p is the Euclid distance of imperialist p to the colony. Then the winning probability of each empire is:

$$W_p = \omega_1 \left| \frac{NTOF_p}{\sum_{i=1}^{N_{imp}} NTOF_i} \right| + \omega_2 \left| \frac{NDis_p}{\sum_{i=1}^{N_{imp}} NDis_i} \right| \quad (27)$$

Thereafter, a random number $rand_p$ is generated between 0 and max{ ω_i } , and empire with the highest value of W_p-$rand_p$ will conquer the colony.

5) REVOLUTION STRATEGY

To increase the diversity of the solution, some weak colonies of proportion R will be replaced by the new ones which obtained by a chaotic sequence based local search strategy. Here the classic Logistic chaotic sequence is used as follows:

$$X_i^{(n+1)} = 4X_i^{(n)}(1\text{-}X_i^{(n)}), i = (1,2,\ldots k) \quad (28)$$

In local search process, the genes of weak countries should be normalized at first. Then a new sequence can be obtained with (28). Genes in the new sequence are sorted in ascending order, and be replaced by their position number. Then we can get a new solution.

IV. COMPUTATIONAL RESULTS AND DISCUSSION

To evaluate the performance of the proposed algorithm, a discrete event simulation model is constructed using Tecnomatix Plant Simulation software shown in FIGURE 2. Dynamic events such as machines breakdown are added into the model. The processing data is obtained from a semiconductor manufacturer in Shanghai, which includes 48 photolithography machines and outputs 350,000 wafers per month. There are three types of wafer lots a, b, and c being processed in the model. At first, the correlation analysis between wafers' remaining cycle time (RCT) and remaining processing layers (RPL) is conducted with the data generated by simulation model. And then the prediction accuracy of deep autoencoder neural network is verified. At last a number of numerical tests are conducted to compare the proposed scheduling algorithm with some other algorithms.

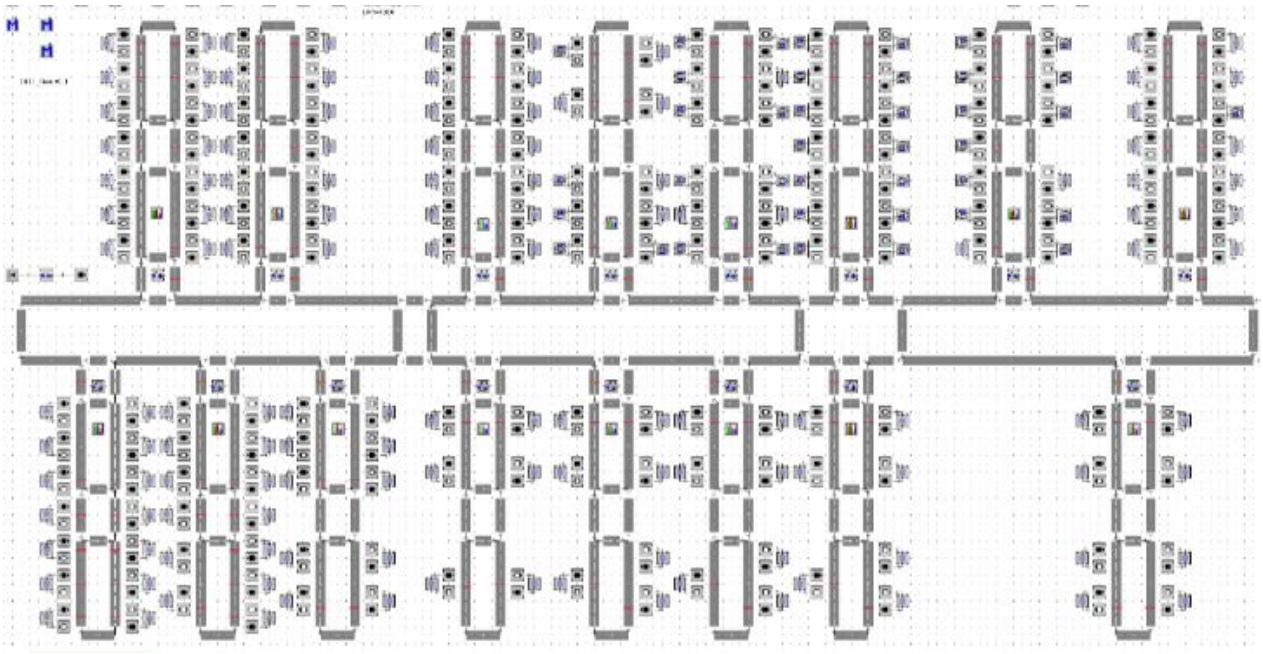


FIGURE 2. **Simulation model of a semiconductor manufacturing system**

A. RCT-RPL Correlation Analysis

We first analyzed the relationship of RCT and RPL with data generated by the simulation model, as shown in FIGURE 3. It is shown that their trend of change is not always keep consistent, especially in the first layer and 12th layer, i.e. the actual completion time of wafers may deviate from the expected completion time. The reason lies with the fact is that different dynamical events may happen in wafer's manufacturing process, and that will affect different wafers in different way, even wafers belonging to the same product. This phenomenon will seriously affect the performance of scheduling algorithm, in which the expected RCT is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

commonly used. Therefore, it is necessary to predict RCT for different wafers and provide it to the scheduling algorithm.

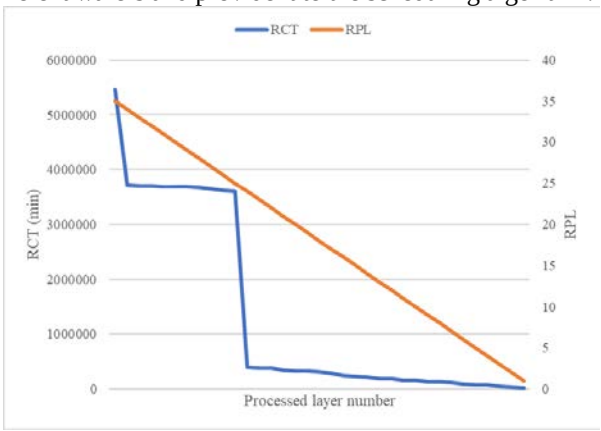


FIGURE 3. **Trend of wafers' RCT and RPL**

B. HYPERPARAMETERS TUNING FOR AUTOENCODERS

Next we built the autoencoders for RCT according to the steps proposed in Section III. The choice space of hyperparameters for autoencoders, however, is rather spacious, making it respectively time-consuming to find an optimized set of parameters. Hence, we searched for the suboptimal solution greedily by selecting the optimal choice for a single hyperparameter while the others were fixed. The order was based on the sequence by which the parameters are introduced in Part A Section III. Take the number of the neurons in the last hidden layer (nNLH) as an exemplification, we selected the best number according to statistical result of the loss function (Formula 17) on the validation set after 30 times of experiments. As is illustrated in FIGURE 4, the mean and variance of the loss function are both at its minimum when nNLH is 18, meaning the bias and variance are both promising under that circumstance. After that, the other parameters, for instance the number of hidden layers, were decided by fixing nNLH and adjusting the value of the pending parameter to achieve the best performance on the validation set.

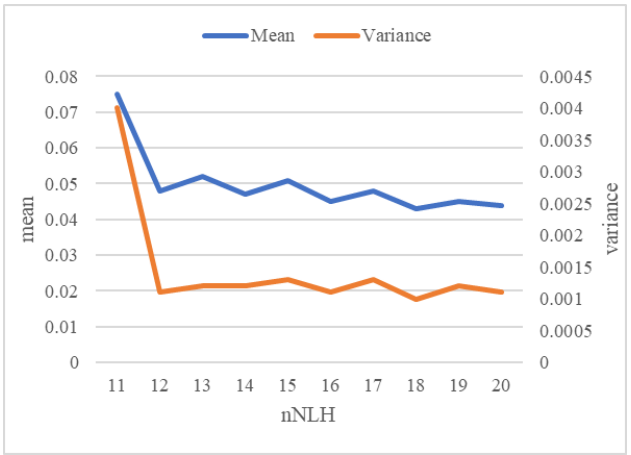


Figure 4. The statistical result of loss function with different nNLHs

The final choices of hyperparameters are demonstrated in TABLE 2, where nHLay denotes the number of hidden layers, lRate denotes the learning rate, nEpoch denotes the number of epochs, Gen denotes the value for generalization penalty, optAl denotes the optimization algorithm and Mome denotes the value of momentum in Adams.

Table 2. The hyperparameters of autoencoders

nHLay	nNLH		lRate
5	[288,144,72,36,18]		0.001
nEpoch	Gen	optAl	Mome
200	500	Adams	0.95

C. Prediction Accuracy Verification

We compared the performance of deep autoencoder neural network based prediction model with three algorithms in the simulation model, namely SVR [25], MLP [25] and GBR [25], as shown in FIGURE 5. We can observe that the prediction proposed perform the best and the prediction accuracy is gradually improving with the increasing of wafers' processed layers, and the average prediction accuracy can be 92%. The predicted RPT is closer to the actual RPT compared to the expected RPT.

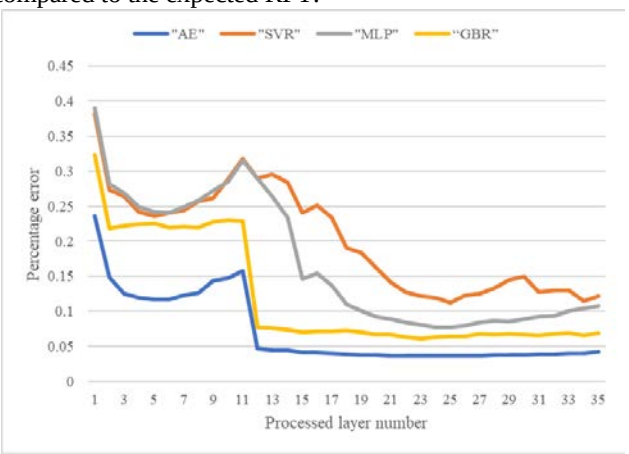


FIGURE 5. **RPT prediction accuracy comparison results in four algorithms**

D. Numerical Tests on Scheduling Algorithm

In this section, we compared the performance of the proposed scheduling algorithm against several approach including ICA [8], GA [29], MMAS [30], SPT [24], SRPT [31] and the standard commercial solver ILOG CPLEX which are selected among the best recent or most relevant algorithms in our scheduling environment. All algorithms will be compared with and without the rolling horizon strategy, as shown in TABLE 3. It is noticeable that all of the algorithm are programmed by Microsoft C# and run on a PC with 2.50GHz Intel Core i5 and 16 GB of RAM memory.

TABLE 3. Description of candidate scheduling Algorithms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2878414, IEEE Access

Algorithm	Description	Algorithm	Description
PRIICA	Improve ICA algorithm with remaining cycle time prediction	RCPLEX	CPLEX with rolling horizon strategy with a time limit of 50 seconds.
RICA	ICA algorithm with rolling horizon strategy	ICA	ICA algorithm used when one machine become idle
RGA	GA algorithm with rolling horizon strategy	GA	GA algorithm used when one machine become idle
RMMAS	MMAS algorithm with rolling horizon strategy	MMAS	MMAS algorithm used when one machine become idle
RSPT	SPT rule with rolling horizon strategy	SPT	SPT rule used when one machine become idle
RSRPT	SRPT rule with rolling horizon strategy	SRPT	SRPT rule used when one machine become idle

TABLE 4. Parameter settings of the improved ICA algorithm

N_{pop}	N_{imp}	ζ	w_1	w_2	T_0
60	2	0.1	1	1.9	1260

According to [8], the parameter settings of PRIICA is shown in Table 4. Each of these algorithms has been running ten times for each case to obtain averaged value, and Error is given by (29). Table 5 shows the comparison results. It can be observed that metaheuristic algorithms with rolling horizon strategy perform better than dispatching rules. Specially, the proposed PRIICA algorithm perform the best among all metaheuristic algorithms with acceptable running time. While in the case without rolling horizon strategy, dispatching rules perform better. It is because that dispatching rules' efficiency remains poor due to lack of a global view, while with a rapid response speed to dynamic arrival wafers; and metaheuristic algorithms can achieve a good global optimization performance within the framework of rolling horizon strategy. However, if metaheuristic algorithms are used immediately when machine becomes idle, then the algorithms will be executed frequently and some lots in the solutions will be repeatedly scheduled, and the globally optimized solutions will be destroyed. On the whole, the improved ICA algorithm with predicted RCT (PRIICA) can greatly reduce wafers' average cycle time.

$$Error = \frac{\mathrm{Alg}_i - \mathrm{Alg}_{best}}{\mathrm{Alg}_{best}} \tag{29}$$

TABLE 5. Total completion time comparison of Algorithms

Methods	One month		Two months		Three months		Avg. run time (s)
	$\sum C$	Error	$\sum C$	Error	$\sum C$	Error	
PRIICA	3898388	0.00%	11042785	0.00%	18648632	0.00%	135
RICA	3961542	1.62%	11210636	1.52%	18926496	1.49%	96
RGA	4196543	7.65%	12176392	10.27%	20498576	9.92%	49
RMMAS	3992675	2.42%	11429674	3.50%	19293874	3.46%	216
RSPT	4470204	14.67%	13122049	18.83%	22199331	19.04%	0.6
RSRPT	4387408	12.54%	12849631	16.36%	21734980	16.55%	0.6
RCPLEX	4264590	9.39%	12376487	12.08%	21013278	12.68%	50
ICA	4610443	18.26%	12541338	13.57%	21660386	16.15%	94
GA	4702625	20.63%	13050363	18.18%	22184412	18.96%	49
MMAS	4550198	16.72%	12790857	15.83%	21729386	16.52%	215
SPT	4352546	11.65%	12673624	14.77%	21162467	13.48%	1.2
SRPT	4244000	8.87%	12426990	12.53%	20711170	11.06%	1.3

V. CONCLUSIONS

This paper deals with the photolithography machines scheduling problem with the objective of total completion

time minimization. Considering the dynamical manufacturing environment and remaining cycle time plays an important role in the scheduling minimizing total completion time, a deep autoencoder neural network is developed to predict wafers' remaining cycle time. And then an improved ICA algorithm in the framework of rolling horizon strategy is used to schedule the dynamical arriving wafers. The method is finally tested in a simulation model. The average RCT prediction accuracy can reach 92%, and wafers' cycle time is significantly decreased. However, there are usually several objectives, such as average cycle time, cycle time variance, throughput rate and on time delivery rate, should be optimized simultaneously in semiconductor manufacturing system. Therefore, in our future work, we will develop a multi-objective scheduling algorithm for photolithography machines.

REFERENCES

[1] S. Kim, S. H. Yea, B. Kim, "Shift scheduling for steppers in the semiconductor wafer fabrication process," . *Iie Transactions*, vol.34, no. 2, pp. 167-177, 2002.

[2] E. Cakici, S. J. Mason. "Parallel machine scheduling subject to auxiliary resource constraints". *Production Planning and Control*, vol. 18, no. 3, pp. 217-225, 2007.

[3] A. Shr, A. Liu, P. P. Chen, "A load balancing method for dedicated photolithography machine constraint," in *Information Technology for Balanced Manufacturing Systems*. Springer, Boston, MA, pp. 339-348, 2006,.

[4] B. Zhou, X. Li, R. Y. K. Fung. "Dynamic scheduling of photolithography process based on Kohonen neural network". *J Intell Manuf*, vol. 26, no. 1, pp. 73-85, 2015.

[5] A. M. Ham, M. Cho. "A practical two-phase approach to scheduling of photolithography production". *IEEE Trans Semicond Manuf*, vol.28, no. 3, pp. 367-373, 2015.

[6] B. Zhou. "An effective estimation of distribution algorithm for parallel litho machine scheduling with reticle constraints". *High Technol Letters*, vol. 22, no. 1, pp. 47-54, 2016.

[7] A. Bitar, S. Dauzère-Pérès, C. Yugma, *et al*. "A memetic algorithm to solve an unrelated parallel machine scheduling problem with auxiliary resources in semiconductor manufacturing". *J. Scheduling*, vol. 19, no. 4, pp. 367-376, 2016.

[8] P. Zhang, Y. Lv, J. Zhang. "An improved imperialist competitive algorithm based photolithography machines scheduling". *Int J Prod Res*, vol. 56, no. 3, pp. 1017-1029, 2018.

[9] S. Leonardi, D. Raz. "Approximating total flow time on parallel machines". *J. Comput. Syst. Sci.*, vol. 73, no. 6, pp. 875-891, 2007.

[10] B. Ankenman, B. L. Nelson, M. Tongarlak, *et al*. "Cycle time prediction for semiconductor manufacturing via simulation on demand". *Working Paper, Northwestern University, Arizona State University, West Virginia University, Evanston*, 2007.

[11] Y. Meidan, B. Lerner, M. Hassoun, *et al*. "Data Mining for Cycle Time Key Factor Identification and Prediction in Semiconductor Manufacturing". *IFAC Proc. Vol.*, vol. 42, no. 4, pp. 217-222, 2009.

[12] J. Wang, J. Zhang. "Big data analytics for forecasting cycle time in semiconductor wafer fabrication system". *Int J Prod Res*, vol. 54, no. 23, pp. 7231-7244, 2016.

[13] T. Chen, Y. C. Wang. "A nonlinearly normalized back propagation network and cloud computing approach for determining cycle time allowance during wafer fabrication". *Rob Comput Integr Manuf*, vol.45, pp. 144-156, 2017.

[14] Y. F. Hung, C. B. Chang. "Dispatching rules using flow time predictions for semiconductor wafer fabrications". *J. Chin. Inst. Ind. Eng.*, vol. 19, no. 1, pp. 67-75, 2002.

[15] T. Chen, Y. C. Wang. "A nonlinear scheduling rule incorporating fuzzy-neural remaining cycle time estimator for scheduling a semiconductor manufacturing factory—a simulation study". *Int J Adv Manuf Technol*, vol. 45, no. 1-2, pp. 110-121, 2009.

[16] L. Li, Q. Yu. "Scheduling strategy of semiconductor production lines with remaining cycle time prediction", in *Proc. Winter Simul. Conf.*, Las Vegas, NV, United states, pp. 3679-3690, 2017.

[17] H. Kamyshanska, R. Memisevic. "The potential energy of an autoencoder". *IEEE Trans Pattern Anal Mach Intell*, vol. 37, no. 6, pp. 1261-1273, 2015.

[18] E. N. Alkhanak, S. P. Lee. "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing". *Future Gener Comput Syst*, vol. 86, pp. 480-506, 2018.

[19] D. Lei, M. Li, L. Wang. "A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold". *IEEE Trans. Cybern.*, 2018.

[20] M. Yazdani, S. M. Khalili, F. Jolai. "A parallel machine scheduling problem with two-agent and tool change activities: an efficient hybrid metaheuristic algorithm". *Int J Computer Integr Manuf*, vol. 29, no. 10, pp. 1075-1088, 2016.

[21] Z. Pan, D. Lei, Q. Zhang. "A New Imperialist Competitive Algorithm for Multiobjective Low Carbon Parallel Machines Scheduling". *Math. Probl. Eng.*, 2018.

[22] M. Zandieh, A. R. Khatami, S. H. A. Rahmati. "Flexible job shop scheduling under condition-based maintenance: improved version of imperialist competitive algorithm". *Appl. Soft Comput. J.*, vol 58, pp. 449-464, 2017.

[23] M. Yazdani, A. Aleti, S. M. Khalili, *et al*. "Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem". *Comput Ind Eng*, vol. 107, pp. 12-24, 2017.

[24] M. L. Pinedo. "Parallel Machine Models (Deterministic)" in *Scheduling: theory, algorithms, and systems*, New York: Springer, 2016.

[25] I. Goodfellow, Y. Bengio, A. Courville, et al. "Optimization for Training Deep Models" in *Deep Learning*, Cambridge: MIT press, pp. 322-325, 2016.

[26] Y. H. Lee, J. Park, S. Kim. "Experimental study on input and bottleneck scheduling for a semiconductor fabrication line[J]. IIE transactions", vol 34, no. 2, pp. 179-190, 2002.

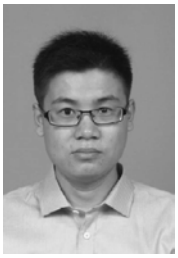
[27] E. Atashpaz-Gargari, C. Lucas. "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition" in *IEEE Congr. Evol. Comput. CEC*, Singapore, pp. 4661-4667, 2007.

[28] M. Rabiee, R. S. Rad, M. Mazinani, *et al*. "An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines". *Int J Adv Manuf Technol*, vol. 71, no. 5-8, pp. 1229-1245, 2014.

[29] E. Vallada, R. Ruiz. "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times". *Eur J Oper Res*, vol. 211, no. 3, pp. 612-622, 2011.

[30] J. P. Arnaout, G. Rabadi, R. Musa. "A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times". *J Intell Manuf*, vol. 21, no. 6, pp. 693-701, 2010.

[31] S. Leonardi, D. Raz. "Approximating total flow time on parallel machines". *J. Comput. Syst. Sci.*,vol. 73, no. 6, pp. 875-891, 2007.



Peng Zhang was born in Xiantao, Hubei Province, China in 1988. He received the B.S. degree in mechanical science and engineering from Huazhong University of Science and Technology, China, in 2011. He is currently pursuing the PH.D. degree in

mechanical engineering at Shanghai Jiaotong University, China. His currrent research interest includes the modeling techniques and scheduling algorithms in complex manufacturing system.

Xinming Zhao received her Master degree in internal conbustion engine in 1989 from Jiangsu University in Jiangsu Province, China. She is now the associate professor of the Institute of Design and Control Engineering for Heavy Equipment at Shanghai Jiaotong University, China. Her main research field includes mechanical design and graphic processing technology, engineering visiualizationa and motion Simulation.

Xia Sheng was born in Nanjing, Jiangsu Province, China in 1993. He received the B.S. degree in mechanical engineering from Nanjing University of Aeronautics and Astronautics, China, in 2015. He is currently pursuing the M.S. degree in mechanical engineering at Shanghai Jiaotong University, China. From 2016 to 2018, he was a Research Assistant with the Intelligent Manufacturing & Information Engineering Institute, Shanghai, China. His research interest includes the modeling techniques and big data analytics of complicated manufacturing system based on artificial intellegence algorithms

Jie Zhang received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics, China, in 1997. She is the Dean of the College of Mechanical Engineering, Donghua University, China. She was with the Institute of Intelligent Manufacturing and Information Engineering, Shanghai Jiao Tong University, China. Her research interests include industrial big data analysis, intelligent production scheduling, production control in intelligent manufacturing system, and intelligent quality analysis.