Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2012, Article ID 404806, 12 pages
doi:10.1155/2012/404806

*Research Article*

# Enhancing Scheduling Performance for a Wafer Fabrication Factory: The Biobjective Slack-Diversifying Nonlinear Fluctuation-Smoothing Rule

**Toly Chen and Yu Cheng Wang**

*Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung 40724, Taiwan*

Correspondence should be addressed to Toly Chen, tcchen@fcu.edu.tw

Received 15 March 2012; Revised 13 June 2012; Accepted 21 June 2012

Academic Editor: W. J. Chris Zhang

A biobjective slack-diversifying nonlinear fluctuation-smoothing rule (biSDNFS) is proposed in the present work to improve the scheduling performance of a wafer fabrication factory. This rule was derived from a one-factor bi-objective nonlinear fluctuation-smoothing rule (1f-biNFS) by dynamically maximizing the standard deviation of the slack, which has been shown to benefit scheduling performance by several previous studies. The efficacy of the biSDNFS was validated with a simulated case; evidence was found to support its effectiveness. We also suggested several directions in which it can be exploited in the future.

## 1. Introduction

Semiconductor manufacturing is undoubtedly one of the most noticeable high-technology industries because semiconductor products have widespread applications. However, the life cycles of new semiconductor products are getting shorter. Therefore, semiconductor manufacturers are facing pressure to meet the various needs of customers within shorter time spans. Manufacturers consider rapid product development, agile production, shortened response times, and similar strategies to be viable. All of these strategies compress the cycle times of related processes. Of the various types of cycle times, production cycle time is particularly important because it determines the time of delivery to customers. In other words, if the production cycle time is shortened, the delivery to customers will be faster. To this end, shortening the production cycle time through effective job dispatching is an important task [1]. Much research has been done concerning semiconductor shop floor control as a special type of supervisory control [2], particularly in the domains of deterministic scheduling and job dispatching. However, Chen and Lin [3], Chen and Wang [4], and Chen [5] have noted that for semiconductor factories, job dispatching is very difficult. Theoretically, this is an NP-hard problem. In practice, many semiconductor factories suffer from lengthy cycle times and thus are not able to make favorable promises to their customers.

This study discusses how to determine the sequence of jobs to be processed on each machine in a semiconductor factory so as to shorten the cycle times of jobs. To this end, an innovative dispatching rule is proposed, which involves the applications of fuzzy logic, artificial neural networks, and mathematical programming.

In this field, some innovative dispatching rules considering job parameters have been proposed recently. For example, Chen [6] reported a nonlinear fluctuation smoothing rule that uses the divisor operator instead of the subtraction operator, which diversifies the slack and makes the nonlinear fluctuation smoothing rule more responsive to changes in the parameters. Chen and Wang [7] also proved that the effects of parameters are balanced better by a nonlinear fluctuation smoothing rule than by a traditional one if the variation in the parameters is large. In short, magnifying the difference in the slack seems to improve scheduling performance. For these reasons, a biobjective slack-diversifying nonlinear fluctuation smoothing rule is presented in this study to improve the scheduling of job dispatching in a wafer fabrication factory.

In a fluctuation smoothing rule, jobs that are expected to have long remaining cycle times are assigned lower slack values, which gives these jobs higher priorities to be processed and quickens their progress. There are two sorts of jobs with long remaining cycle times. The first sort comprises jobs that are just in their early stages; these jobs still have many stages to undergo. It is not necessary to deal with jobs of this type. The second type comprises jobs that have been delayed for long periods of time; these jobs have undergone few stages and have more unprocessed stages than the other jobs that started at the same time. Such a situation should be tackled somehow. However, even though these jobs have high slack values according to a fluctuation smoothing rule, they might not be assigned appropriately high priorities because sometimes many jobs have high slack values at the same time and we are not able to determine an absolute sorting for these jobs. To tackle this problem, we need a rule that is able to generate slack values that are as diverse as possible. To this end, we propose the biobjective slack-diversifying nonlinear fluctuation smoothing rule. This rule differs from 1f-biNFS because it maximizes the difference in the slack as measured by the standard deviation of the slack. There are many factors which must be optimized to achieve this goal, so a complex optimization problem must be solved to produce the rule. We apply a polynomial fitting technique to convert it into a more tractable form, for which several optimal solutions can be found. After screening some values from the specified range, the remaining values are used to construct an optimized 1f-biNFS rule.

The later sections of this paper are arranged in the following way. Section 2 is dedicated to the literature review. Section 3 provides the details of the proposed methodology. In Section 4, a simulated case is used to validate the effectiveness of the biobjective slack-diversifying nonlinear fluctuation smoothing rule. The performance levels of some existing rules in this field are also examined using the simulated data. Section 5 concludes this paper and points out some interesting topics for future work.

## 2. Literature Review

Semiconductor manufacturing can be divided into four stages: wafer fabrication, wafer probing, packaging, and final testing. The most important and most time-consuming stage is wafer fabrication, which starts with approximately 25 wafers grouped as a lot. This lot is passed through hundreds of operations to build up complex layers of patterned metal and wafer materials that produce the required circuitry. In this study, we investigate job dispatching for this stage. Among the various categories of methods (including dispatching rules, heuristics, data mining-based approaches [8, 9], agent technologies ([8, 10–12], and simulation) in this field, dispatching rules (e.g., first-in first-out (FIFO), earliest due date (EDD), least slack (LS), shortest processing time (SPT), shortest remaining processing time (SRPT), critical ratio (CR), the fluctuation smoothing rule for the mean cycle time (FSMCT), the fluctuation smoothing rule for cycle time variation (FSVCT), least total work (LTWK), modified due date (MDD), operation due date (ODD), cost over time (COVERT), FIFO+, SRPT+, and SRPT++) have received a lot of attention these years [8–10] and are also the most prevalent method in practical applications. For the details of the traditional dispatching rules, refer to Lu et al. [13]. A recent simulation comparison is presented in Chiang and Fu [14].

Some advances in this field are introduced in the following. Altendorfer et al. [15] proposed the work in parallel queue (WIPQ) rule targeting at maximizing throughput at a low level of work in process (WIP). Zhang et al. [16] proposed the dynamic bottleneck detection (DBD) approach by classifying workstations into several categories and then applying different dispatching rules to these categories. Three dispatching rules including FIFO, the shortest processing time until the next bottleneck (SPNB), and CR were used. Depending on the current conditions in the wafer fabrication factory, Hsieh et al. [9] chose one approach from FSMCT, FSVCT, largest deviation first (LDF), one step ahead (OSA), and FIFO.

Chen [17] modified FSMCT and proposed the nonlinear FSMCT (NFSMCT) rule, in which he smoothed the fluctuation in the estimated remaining cycle time and balanced it with that of the release time or the mean release rate. To diversify the slack, the division operator was applied instead. Followed by Chen [18], the one-factor tailored NFSMCT (1f-TNFSMCT) rule and the one-factor tailored nonlinear FSVCT (1f-TNFSVCT) rule were proposed. Both rules contain an adjustable parameter in order to customize them for a target wafer fabrication factory. As a multiple-objective study, Chen et al. [19] pro-posed a biobjective nonlinear fluctuation smoothing rule with an adjustable factor (1f-biNFS) to optimize the average cycle time and cycle time variation at the same time. More degrees of freedom seem to be conducive to the performance of customizable rules. For this reason, Chen et al. [19] extended 1f-biNFS to a biobjective fluctuation smoothing rule with four adjustable factors (4f-biNFS). For a summary of these rules refer to Table 1. One drawback of them is that only static factors are used, and these factors need to be determined in advance. To this end, most studies (e.g., [17–19]) have performed extensive simulation. Such simulation is not only time consuming but it also fails to consider enough possible combinations of these factors. Chen [6] established a mechanism that was able to adjust factor values for 1f-biNFS dynamically (dynamic 1f-biNFS). However, even though satisfactory results were obtained in that experiment, there was no theoretical basis supporting the proposed mechanism. Chen [20] tried to relate the scheduling performance to the factor values with a back propagation network (BPN). Artificial neural networks have been widely applied to various control fields [21–23]. When such applications work, one can find the factor values that contribute to optimal scheduling performance. However, the explanatory ability of the BPN was not sufficient.

Table 1: The summary of some dispatching rules.

| Rule name | Formula |
|---|---|
| 1f-TNFSMCT | $\mathrm{SK}_{ij} = \left(\dfrac{\beta}{\alpha\left(\mathrm{RCTE}_{ij} - \min\left(\mathrm{RCTE}_{ij}\right)\right)}\right)^{\xi} \cdot (R_i - \mathrm{RCTE}_{ij} + \xi(\mathrm{RCTE}_{ij} - \min(R_i)))$ |
| 1f-TNFSVCT | $\mathrm{SK}_{ij} = \left(\dfrac{\beta\lambda}{\gamma\left(\mathrm{RCTE}_{ij} - \min(\mathrm{RCTE}_{ij})\right)}\right)^{\xi} \cdot \left(\dfrac{i}{\lambda} - \mathrm{RCTE}_{ij} + \left(\mathrm{RCTE}_{ij} - \dfrac{1}{\lambda}\right)\xi\right)$ |
| 1f-biNFS | $\mathrm{SK}_{ij} = \dfrac{((i/\lambda) - (1/\lambda))^{1-\xi}(R_i - \min(R_i))^{\xi}\left(\mathrm{RCTE}_{ij} - \min\left(\mathrm{RCTE}_{ij}\right)\right)}{((N/\lambda) - (1/\lambda))^{1-\xi}(\max(R_i) - \min(R_i))^{\xi}\left(\max\left(\mathrm{RCTE}_{ij}\right) - \min\left(\mathrm{RCTE}_{ij}\right)\right)}$ |
| 4f-biNFS | $\mathrm{SK}_{ij} = \left(R_i - \mathrm{RCTE}_{ij} + \left(\mathrm{RCTE}_{ij} - \min(R_i)\right) \cdot f_1\right) \cdot \alpha^{-f_2}$<br>$\cdot\left(\dfrac{i}{\lambda} - \mathrm{RCTE}_{ij} + \left(\mathrm{RCTE}_{ij} - \dfrac{1}{\lambda}\right) \cdot f_3\right) \cdot \left(\dfrac{\gamma}{\lambda}\right)^{-f_4}$<br>$\cdot\left(\dfrac{(\mathrm{RCTE}_{ij} - \min\left(\mathrm{RCTE}_{ij}\right))}{\beta}\right)^{-(f_2+f_4)}$ |
| Dynamic 1f biNFS | $\mathrm{SK}_{ij} = \dfrac{((i/\lambda) - (1/\lambda))^{1-\xi(t)}(R_i - \min(R_i))^{\xi(t)}\left(\mathrm{RCTE}_{ij} - \min\left(\mathrm{RCTE}_{ij}\right)\right)}{((N/\lambda) - (1/\lambda))^{1-\xi}(\max(R_i) - \min(R_i))^{\xi}\left(\max\left(\mathrm{RCTE}_{ij}\right) - \min\left(\mathrm{RCTE}_{ij}\right)\right)}$<br>$\xi(t) = \left(\dfrac{1}{2}\right)\left(\sin\left(\dfrac{\pi}{c}t\right) + 1\right)$ |

## 3. Methodology

The variables are defined as follows:

(1) $R_i$: the release time of job $i$, $i = 1 \sim N$,

(2) $\mathrm{BQ}_i$: the total queue length before the bottlenecks at $R_i$,

(3) $\mathrm{CT}_i$: the cycle time (actual value) of job $i$,

(4) $\mathrm{CTE}_i$: the estimated cycle time of job $i$,

(5) $D_i^{(l)}$: the delay of the $l$th recently completed job at $R_i$, $l = 1 \sim 3$,

(6) $\mathrm{FQ}_i$: the total queue length in the whole factory at $R_i$,

(7) $Q_i$: the total queue length on the processing route of job $i$ at $R_i$,

(8) $\mathrm{RCT}_{ij}$: the remaining cycle time (actual value) of job $i$ since step $j$,

(9) $\mathrm{RCTE}_{ij}$: the estimated remaining cycle time of job $i$ since step $j$,

(10) $\mathrm{SCT}_{ij}$: the step cycle time (actual value) of job $i$ until step $j$,

(11) $\mathrm{SCTE}_{ij}$: the estimated step cycle time of job $i$ until step $j$,

(12) $\mathrm{WIP}_i$: the factory WIP at $R_i$,

(13) $\mathrm{SK}_{ij}$: the slack of job $i$ at step $j$,

(14) $U_i$: the average factory utilization at $R_i$,

(15) $\alpha$: $\max(R_i) - \min(R_i)$,

(16) $\beta$: $\max(\mathrm{RCTE}_{ij}) - \min(\mathrm{RCTE}_{ij})$,

(17) $\gamma$: $N - 1$,

(18) $\lambda$: the mean release rate.

Obviously,

$$\mathrm{CT}_i = \mathrm{SCT}_{ij} + \mathrm{RCT}_{ij}. \tag{1}$$

Replacing all variables with their estimates gives

$$\mathrm{CTE}_i = \mathrm{SCTE}_{ij} + \mathrm{RCTE}_{ij}. \tag{2}$$

*3.1. Remaining Cycle Time Estimation.* Before applying the biobjective slack-diversifying nonlinear fluctuation smoothing rule, the remaining cycle time required for each job must be estimated in advance. There is not a great deal of research in this field, but the fuzzy c-means (FCM) and fuzzy back propagation network (FBPN) approach of Chen et al. [24] has been shown to be effective [25–27] and therefore has been used in this study. In the FCM-FBPN approach, FCM is first used to cluster jobs with similar attributes. FCM performs classification by minimizing the following objective function:

$$\mathrm{Min} \sum_{k=1}^{K} \sum_{i=1}^{n} \mu_{i(k)}^{m} e_{i(k)}^{2}, \tag{3}$$

where $K$ is the required number of categories; $n$ is the number of jobs; $\mu_{i(k)}$ represents the membership of job $i$ belonging to category $k$; $e_{i(k)}$ measures the distance from job $i$ to the centroid of category $k$; $m \in [1, \infty)$ is a parameter to increase or decrease the fuzziness. The procedure of applying FCM to classify jobs is as follows.

(1) Establish an initial classification result.

(2) Iterations: obtain the centroid of each category as

$$\begin{aligned}
\overline{x}_{(k)} &= \left\{\overline{x}_{(k)j}\right\}, \\
\overline{x}_{(k)j} &= \frac{\sum_{i=1}^{n} \mu_{i(k)}^{m} x_{ij}}{\sum_{i=1}^{n} \mu_{i(k)}^{m}}, \\
\mu_{i(k)} &= \frac{1}{\sum_{l=1}^{K} \left(e_{i(k)}/e_{i(l)}\right)^{2/(m-1)}}, \\
e_{i(k)} &= \sqrt{\sum_{\text{all } j} \left(x_{ij} - \overline{x}_{(k)j}\right)^{2}},
\end{aligned} \tag{4}$$

where $\overline{x}_{(k)}$ is the centroid of category $k$ and $\mu_{i(k)}^{(t)}$ is the membership of job $i$ belonging to category $k$ after the $t$th iteration.

(3) Remeasure the distance of each job to the centroid of every category, and then recalculate the corresponding membership.

(4) Stop if the following condition is satisfied. Otherwise, return to step (2):

$$\max_{k} \max_{i} \left| \mu_{i(k)}^{(t)} - \mu_{i(k)}^{(t-1)} \right| < d, \tag{5}$$

where $d$ is a real number representing the threshold of membership convergence.

Finally, the separate distance test ($S$ test) proposed by Xie and Beni [28] can be applied to determine the optimal number of categories $K$:

$$\text{Min } S \tag{6}$$

subject to

$$\begin{aligned}
J_m &= \sum_{k=1}^{K} \sum_{i=1}^{n} \mu_{i(k)}^{m} e_{i(k)}^{2}, \\
e_{\min}^{2} &= \min_{p \neq q} \left( \sum_{\text{all } j} \left(\overline{x}_{(p)j} - \overline{x}_{(q)j}\right)^{2} \right), \\
S &= \frac{J_m}{n \times e_{\min}^{2}}, \\
K &\in Z^{+}.
\end{aligned} \tag{7}$$

The $K$ value minimizing $S$ determines the optimal number of categories.

The remaining cycle time of a job that is being processed in a wafer fabrication factory is the time still required to complete the job. If the job has just been released into the wafer fabrication factory, then the remaining cycle time of the job is its cycle time. The remaining cycle time is an important performance measure for all work-in-progress (WIP) in a wafer fabrication factory. To predict the remaining cycle time, we usually subtract the step cycle time from the cycle time forecast:

$$\text{RCTE}_{nj} = \text{CTE}_{nj} - \text{SCT}_{nj}. \tag{8}$$

For this reason, we need to predict both the cycle time and the step cycle time.

After clustering, a portion of the jobs in each category is fed back into the FBPN as "training examples" in order to determine the parameter values for the category. The configuration of the FBPN is as follows.

(1) Inputs: eight parameters are associated with the $n$th example/job including $U_n$, $Q_n$, $\text{BQ}_n$, $\text{FQ}_n$, $\text{WIP}_n$, and $D_n^{(i)}$ ($i = 1 \sim 3$).

(2) There is a single hidden layer.

(3) The number of neurons in the hidden layer is the same as the number of neurons in the input layer.

(4) Output: the estimated (normalized) cycle time ($\text{CTE}_n$) or estimated step cycle time ($\text{SCTE}_{nj}$) of the example. In other words, there are two groups of BPNs. The first group estimates the $\text{CTE}_n$'s of all the jobs to be scheduled, while the other group estimates their $\text{SCTE}_{nj}$'s. The remaining cycle time estimate ($\text{RCTE}_{nj}$) can be derived by subtracting $\text{SCTE}_{nj}$ from $\text{CTE}_n$.

(5) The network learning rule is the Delta rule.

(6) The transformation function is the Sigmoid function

$$f(x) = \frac{1}{(1 + e^{-x})}. \tag{9}$$

(7) The learning rate ($\eta$) ranges from 0.01 to 1.0.

(8) Initial conditions: because FBPNs tend to be very sensitive to initial conditions, in this study, a GA is employed to generate the initial values of the connection weights in the FBPN. Each chromosome is a vector of about 132 connection weights (see Figure 1). The connection weights are read off the FBPN and placed in a vector from left to right and from top to bottom. Each gene in the chromosome is a real number instead of a bit. To calculate the fitness of a given chromosome, the connection weights in the chromosome are assigned to the corresponding connections in the FBPN, the FBPN is trained using the training data, and the RMSE is returned. A low RSME value indicates high fitness:

$$f = \frac{1000}{\text{RMSE}}. \tag{10}$$

An initial population of 100 vectors is chosen randomly, with each connection weight set to some uniformly distributed random value between −1.0 and +1.0. The mutation operator selects n noninput neurons and, for each incoming connection to those neurons, adds a uniformly distributed random value between −1.0 and +1.0 to the connection weight. The crossover operator takes two parent connection weight vectors; each noninput neuron in the offspring vector selects one of the parents randomly and

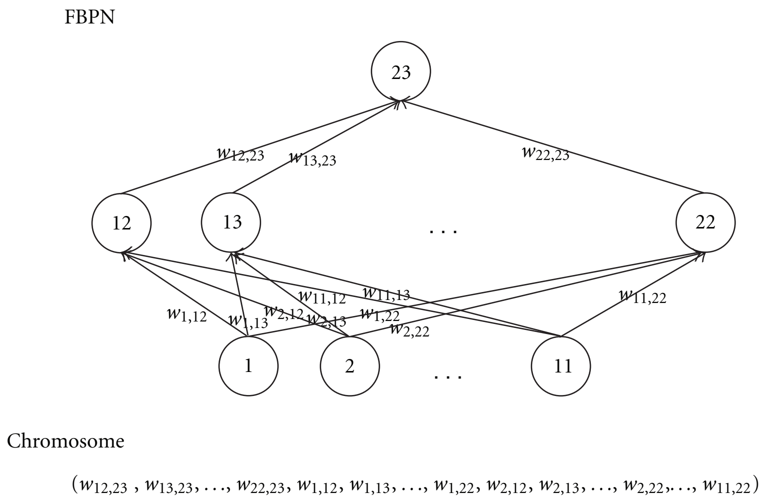


Figure 1: The chromosome used in the FCM-GA-FBPN.

copies the connection weights on the incoming connections from that parent to the offspring. Only one offspring is generated. In the child network, the weights of the incoming connections to neurons 12 and 23 come from parent 1, while those of the incoming connections to neurons 13 to 22 come from parent 2.

(9) Batch learning: the procedure for determining the parameter values is as follows. After preclassification, some of the adopted examples in each category are fed into the FBPN as "training examples" to determine the parameter values for the category. Two phases are involved at the training stage: the forward phase and the backward phase. In the forward phase, inputs are multiplied with weights, summed, and transferred to the hidden layer. Subsequently, activated signals are output from the hidden layer as

$$\tilde{h}_j = \frac{1}{1+e^{-\tilde{n}_j^h}}, \tag{11}$$

where

$$\begin{aligned} \tilde{n}_j^h &= \tilde{I}_j^h(-)\tilde{\theta}_j^h, \\ \tilde{I}_j^h &= \sum_{\text{all } j} \tilde{w}_{ij}^h(\times)\tilde{x}_{(i)}. \end{aligned} \tag{12}$$

$\tilde{h}_j$ values are also transferred to the output layer with the same procedure. Finally, the output of the FBPN is generated as

$$\tilde{o} = \frac{1}{1+e^{-\tilde{n}^o}}, \tag{13}$$

where

$$\begin{aligned} \tilde{n}^o &= \tilde{I}^o(-)\tilde{\theta}^o, \\ \tilde{I}^o &= \sum_{\text{all } j} \tilde{w}_j^o(\times)\tilde{h}_j. \end{aligned} \tag{14}$$

To improve the applicability of the FBPN and to facilitate comparisons with conventional techniques, the fuzzy-valued output $\tilde{o}$ is defuzzified according to the following formula:

$$d(\tilde{o}) = \int_0^1 E(o^\alpha)d\alpha, \tag{15}$$

where $o^\alpha$ is the $\alpha$ cut of $\tilde{o}$. Then the output $o$ is compared with the normalized actual cycle time (or step cycle time) $a$, for which the RMSE is calculated as

$$\text{RMSE} = \sqrt{\frac{\sum_{\text{all trained examples}} (o-a)^2}{\text{number of trained examples}}}. \tag{16}$$

In the backward phase, the deviation between $o$ and $a$ is propagated backward, and the error terms of neurons in the output and hidden layers can be calculated, respectively, as

$$\begin{aligned} \delta^o &= o(1-o)(a-o), \\ \tilde{\delta}_j^h &= \tilde{h}_j(\times)\left(1-\tilde{h}_j\right)(\times)\tilde{w}_j^o\delta^o. \end{aligned} \tag{17}$$

Based on these error terms, adjustments to be made for connecting weights and thresholds can be obtained as

$$\begin{aligned} \Delta\tilde{w}_j^o &= \eta\delta^o\tilde{h}_j, \\ \Delta\tilde{w}_{ij}^h &= \eta\tilde{\delta}_j^h(\times)\tilde{x}_i, \\ \Delta\theta^o &= -\eta\delta^o, \\ \Delta\tilde{\theta}_j^h &= -\eta\tilde{\delta}_j^h. \end{aligned} \tag{18}$$

It is based on the basic gradient descent algorithm. For details refer to Chen [29] and Pendharkar [30]. To accelerate convergence, a momentum term can be added to the learning expressions. For example,

$$\Delta\tilde{w}_j^o = \eta\delta^o\tilde{h}_j + \alpha\left(\tilde{w}_j^o(t) - \tilde{w}_j^o(t-1)\right). \tag{19}$$

Theoretically, network learning stops when the RMSE falls below a prespecified level, or when the improvement in the RMSE becomes negligible over several epochs, or when a large number of epochs have already been run. Then test examples are fed into the FBPN and the accuracy of the network is measured with the RMSE. However, the accumulation of fuzziness during the training process continuously increases the lower bound, the upper bound, and the spread of the fuzzy-valued output $\tilde{o}$ (and those of many other fuzzy parameters); this might prevent the RMSE (calculated with the defuzzified output $o$) from converging to its minimal value. Conversely, network learning tends to shrink the centers of some fuzzy parameters. A fuzzy parameter can become invalid if its lower bound is higher than its center. To deal with this problem, the lower and upper bounds of all fuzzy numbers in the FBPN will no longer be modified if the following index converges to a minimal value:

$$\alpha\sqrt{\frac{\sum_{\text{all examples}}\min\left((o_1-a)^2,(o_3-a)^2\right)}{\text{number of examples}}} + (1-\alpha)\sqrt{\frac{\sum_{\text{all examples}}\max\left((o_1-a)^2,(o_3-a)^2\right)}{\text{number of examples}}}, \quad 0<\alpha<1. \tag{20}$$

Finally, the FBPN can be applied to estimate the cycle time or the step cycle time of a new job. When a new job is released into the factory, the eight parameters associated with the new job are recorded. Then the FBPN is applied to estimate the cycle time or step cycle time of the new job.

*3.2. The Bicriteria Slack-Diversifying Nonlinear Fluctuation Smoothing Rule.* The bicriteria slack-diversifying nonlinear fluctuation smoothing rule is derived by diversifying the slack in the 1f-biNFS rule:

$$\begin{aligned}\mathrm{SK}_{ij} &= \frac{(i/\lambda-1/\lambda)^{1-\xi}(R_i-\mathcal{B})^{\xi}\left(\mathrm{RCTE}_{ij}-\mathcal{A}\right)}{(N/\lambda-1/\lambda)^{1-\xi}(\max(R_i)-\mathcal{B})^{\xi}(\mathcal{C}-\mathcal{A})}\\ &= a_i^{1-\xi}b_i^{\xi}c_{ij}\\ &= a_ic_{ij}\left(\frac{b_i}{a_i}\right)^{\xi},\end{aligned} \tag{21}$$

where, $\mathcal{A}=\min(\mathrm{RCTE}_{ij})$, $\mathcal{B}=\min(R_i)$, $\mathcal{C}=\max(\mathrm{RCTE}_{ij})$

$$\begin{aligned}a_i &= \frac{(i/\lambda)-(1/\lambda)}{(N/\lambda)-(1/\lambda)},\\ b_i &= \frac{R_i-\min(R_i)}{\max(R_i)-\min(R_i)},\\ c_{ij} &= \frac{\mathrm{RCTE}_{ij}-\min\left(\mathrm{RCTE}_{ij}\right)}{\max\left(\mathrm{RCTE}_{ij}\right)-\min\left(\mathrm{RCTE}_{ij}\right)}.\end{aligned} \tag{22}$$

The following two theorems explain the theoretical properties of 1f-biNFS.

**Theorem 1.** *1f-biNFS is more responsive than the traditional fluctuation smoothing rules to changes in $R_n$ if $RCTE_{nj}R_n$ [6].*

**Theorem 2.** *The effects of parameters are balanced better by 1f-biNFS than by the traditional fluctuation smoothing rules if $RCTE_{nj}-\min(RCTE_{nj})\geq R_n-\min(R_n)$, that is, if the variation in $RCTE_{nj}$ is greater than that in $R_n$, which is a common phenomenon in a wafer fabrication factory [7].*

However, (21) is difficult to deal with. For this reason, the following polynomial fitting technique is used to convert it into a more tractable form:

$$x^{\xi}\cong(0.94+1.77\xi-2.45\xi^2)+(0.02-0.16\xi+1.01\xi^2)x. \tag{23}$$

The mean absolute percentage error (RMSE) of (23) is less than 5% when $x\leq 20$. The RMSE will not be a serious problem since it is the $\xi$ value associated with the minimum $\sigma_{\mathrm{SK}_{ij}}$ to be found, not the $\mathrm{SK}_{ij}$ values. Such a polynomial fitting technique is especially effective when $x$ exceeds 1 (see Figure 2). Applying (23) to (21) yields

$$\begin{aligned}\mathrm{SK}_{ij} &\cong a_ic_{ij}\left(0.94+1.77\xi-2.45\xi^2+(0.02-0.16\xi+1.01\xi^2)\frac{b_i}{a_i}\right)\\ &= \left(0.94a_ic_{ij}+0.02b_ic_{ij}\right)+\left(1.77a_ic_{ij}-0.16b_ic_{ij}\right)\xi\\ &\quad+\left(-2.45a_ic_{ij}+1.01b_ic_{ij}\right)\xi^2\\ &= d_{ij}+e_{ij}\xi+f_{ij}\xi^2,\end{aligned} \tag{24}$$

where

$$\begin{aligned}d_{ij} &= 0.94a_ic_{ij}+0.02b_ic_{ij},\\ e_{ij} &= 1.77a_ic_{ij}-0.16b_ic_{ij},\\ f_{ij} &= -2.45a_ic_{ij}+1.01b_ic_{ij}.\end{aligned} \tag{25}$$

To diversify the slack, the standard deviation of the slack is to be maximized:

$$\begin{aligned}\sigma_{\mathrm{SK}_{ij}} &= \sqrt{\frac{\sum_{i=1}^{N}\left(\mathrm{SK}_{ij}-\overline{\mathrm{SK}_j}\right)^2}{N-1}}\\ &= \sqrt{\frac{1}{\gamma}}\sqrt{\sum_{i=1}^{N}\mathrm{SK}_{ij}^2-\frac{1}{N}\left(\sum_{i=1}^{N}\mathrm{SK}_{ij}\right)^2}.\end{aligned} \tag{26}$$

It is equivalent to maximizing the following term:

$$\begin{aligned}&\sum_{i=1}^{N}\mathrm{SK}_{ij}^2-\frac{1}{N}\left(\sum_{i=1}^{N}\mathrm{SK}_{ij}\right)^2\\ &\quad=\sum_{i=1}^{N}\left(d_{ij}+e_{ij}\xi+f_{ij}\xi^2\right)^2-\frac{1}{N}\left(\sum_{i=1}^{N}\left(d_{ij}+e_{ij}\xi+f_{ij}\xi^2\right)\right)^2\end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^{N}\left(d_{ij}^2 + e_{ij}^2\xi^2 + f_{ij}^2\xi^4 + 2d_{ij}e_{ij}\xi + 2d_{ij}f_{ij}\xi^2 + 2e_{ij}f_{ij}\xi^3\right)\\
&\quad - \frac{1}{N}\left(\left(\sum_{i=1}^{N}d_{ij}\right)^2 + \left(\sum_{i=1}^{N}e_{ij}\right)^2\xi^2 + \left(\sum_{i=1}^{N}f_{ij}\right)^2\xi^4\right.\\
&\qquad\left. + 2\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}e_{ij}\xi + 2\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}f_{ij}\xi^2 + 2\sum_{i=1}^{N}e_{ij}\sum_{i=1}^{N}f_{ij}\xi^3\right)\\
&= \sum_{i=1}^{N}d_{ij}^2 - \frac{1}{N}\left(\sum_{i=1}^{N}d_{ij}\right)^2 + \left(2\sum_{i=1}^{N}d_{ij}e_{ij} - \frac{2}{N}\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}e_{ij}\right)\xi\\
&\quad + \left(\sum_{i=1}^{N}e_{ij}^2 - \frac{1}{N}\left(\sum_{i=1}^{N}e_{ij}\right)^2 + 2\sum_{i=1}^{N}d_{ij}f_{ij} - \frac{2}{N}\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}f_{ij}\right)\xi^2\\
&\quad + \left(2\sum_{i=1}^{N}e_{ij}f_{ij} - \frac{2}{N}\sum_{i=1}^{N}e_{ij}\sum_{i=1}^{N}f_{ij}\right)\xi^3\\
&\quad + \left(\sum_{i=1}^{N}f_{ij}^2 - \frac{1}{N}\left(\sum_{i=1}^{N}f_{ij}\right)^2\right)\xi^4.
\end{aligned} \tag{27}$$

Taking the derivative of (27) with respect to $\xi$, and setting it equal to zero, we obtain

$$\begin{aligned}
&\left(2\sum_{i=1}^{N}d_{ij}e_{ij} - \frac{2}{N}\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}e_{ij}\right)\\
&\quad + \left(2\sum_{i=1}^{N}e_{ij}^2 - \frac{2}{N}\left(\sum_{i=1}^{N}e_{ij}\right)^2 + 4\sum_{i=1}^{N}d_{ij}f_{ij} - \frac{4}{N}\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}f_{ij}\right)\xi\\
&\quad + \left(6\sum_{i=1}^{N}e_{ij}f_{ij} - \frac{6}{N}\sum_{i=1}^{N}e_{ij}\sum_{i=1}^{N}f_{ij}\right)\xi^2\\
&\quad + \left(4\sum_{i=1}^{N}f_{ij}^2 - \frac{4}{N}\left(\sum_{i=1}^{N}f_{ij}\right)^2\right)\xi^3\\
&\quad = w_1 + w_2\xi + w_3\xi^2 + w_4\xi^3 = 0,
\end{aligned} \tag{28}$$

where

$$\begin{aligned}
w_1 &= 2\sum_{i=1}^{N}d_{ij}e_{ij} - \frac{2}{N}\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}e_{ij},\\
w_2 &= 2\sum_{i=1}^{N}e_{ij}^2 - \frac{2}{N}\left(\sum_{i=1}^{N}e_{ij}\right)^2 + 4\sum_{i=1}^{N}d_{ij}f_{ij} - \frac{4}{N}\sum_{i=1}^{N}d_{ij}\sum_{i=1}^{N}f_{ij},\\
w_3 &= 6\sum_{i=1}^{N}e_{ij}f_{ij} - \frac{6}{N}\sum_{i=1}^{N}e_{ij}\sum_{i=1}^{N}f_{ij},\\
w_4 &= 4\sum_{i=1}^{N}f_{ij}^2 - \frac{4}{N}\left(\sum_{i=1}^{N}f_{ij}\right)^2.
\end{aligned} \tag{29}$$

The optimal solution $\xi^*$ can be derived as

$$\begin{aligned}
&\xi^*(1)\\
&\quad = \frac{1}{6w_4}\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 + 12\sqrt{3}w4\\
&\qquad\qquad \times (4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3}\\
&\qquad - \frac{2}{3}(3w_2w_4 - w_3^2)/w4\\
&\qquad /\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 + 12\sqrt{3}w_4\\
&\qquad\qquad \times (4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3} - \frac{1}{3}w_3w_4,\\
&\xi^*(2)\\
&\quad = \frac{-1}{12w_4}\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 + 12\sqrt{3}w_4\\
&\qquad\qquad \times \Big(4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3}\\
&\qquad + \frac{1}{3}(3w_2w_4 - w_3^2)/w_4\\
&\qquad /\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 + 12\sqrt{3}w_4\\
&\qquad\qquad \times (4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3} - \frac{w_3}{3w_4}\\
&\qquad + \frac{\sqrt{3}}{2}\Big(\frac{1}{6w_4}(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3\\
&\qquad\qquad + 12\sqrt{3}w_4 \times (4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3}\\
&\qquad\quad + \frac{2}{3}(3w_2w_4 - w_3^2)/w_4\\
&\qquad\quad /\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 + 12\sqrt{3}w_4\\
&\qquad\qquad \times (4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3}\Big)i,
\end{aligned} \tag{30}$$

$$\begin{aligned}
&\xi^*(3)\\
&\quad = \frac{-1}{12w_4}\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 + 12\sqrt{3}w_4\\
&\qquad\qquad \times (4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4\\
&\qquad\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3}
\end{aligned}$$

$$\begin{aligned}
&+\frac{1}{3}(3w_2w_4 - w_3^2)/w_4 \\
&/\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 \\
&\quad + 12\sqrt{3}w_4(4w_2^3w_4 - w_2^2w_3^2 - 18w_1w_2w_3w_4 \\
&\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3} - \frac{w_3}{3w_4} \\
&-\frac{\sqrt{3}}{2}\bigg(\frac{1}{6w_4}(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 \\
&\qquad + 12\sqrt{3}w_4(4w_2^3w_4 \\
&\qquad\qquad - w_2^2w_3^2 - 18w_1w_2w_3w_4 \\
&\qquad\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3} \\
&+\frac{2}{3}(3w_2w_4 - w_3^2)/w_4 \\
&/\Big(36w_2w_3w_4 - 108w_1w_4^2 - 8w_3^3 \\
&\quad + 12\sqrt{3}w4(4w_2^3w4 - w_2^2w_3^2 - 18w_1w_2w_3w_4 \\
&\qquad +27w_1^2w_4^2 + 4w_1w_3^3)^{1/2}\Big)^{1/3}\bigg)i.
\end{aligned} \tag{31}$$

However, not all of these terms are the answer since $\xi^*$ needs to satisfy the following constraints:

$$0 \le \xi^* \le 1 \tag{32}$$

$$\sigma''_{\mathrm{SD}_{ij}}(\xi^*) = w_2 + w_3\xi^* + w_4(\xi^*)^2 \le 0. \tag{33}$$

Further, (31) and (32) are complex numbers that will only be considered if their imaginary parts are equal to zero. An example is given in Table 2 to illustrate the procedure mentioned previously. The optimal solution is $\xi^* = 0.94$ with the maximum $\sigma_{\mathrm{SK}_{ij}}$ equal to 64115.3. Finally, $\xi^*$ can be used to construct an optimized 1f-biNFS as

$$\begin{aligned}
\mathrm{SK}_{ij} &= \left(\frac{\beta}{\alpha\big(\mathrm{RCTE}_{ij} - \min\big(\mathrm{RCTE}_{ij}\big)\big)}\right)^{0.94} \\
&\quad \cdot \Big(R_i - \mathrm{RCTE}_{ij} + 0.94\big(\mathrm{RCTE}_{ij} - \min(R_i)\big)\Big).
\end{aligned} \tag{34}$$

However, it is possible that a job might have a very high or a very low slack value, which could distort the results. For this reason, we exclude the jobs with the highest or lowest slack value from (32):

$$\begin{aligned}
\sigma'_{\mathrm{SK}_{ij}} &= \sqrt{\frac{\sum_{\mathrm{SK}_{ij}\neq Q}\left(\mathrm{SK}_{ij} - \overline{\mathrm{SK}_j}\right)^2}{N-3}}, \\
&= \sqrt{\frac{1}{\gamma-2}}\sqrt{\sum_{\mathrm{SK}_{ij}\neq Q}\mathrm{SK}_{ij}^2 - \frac{1}{N-2}\left(\sum_{\mathrm{SK}_{ij}\neq Q}\mathrm{SK}_{ij}\right)^2},
\end{aligned} \tag{35}$$

where $Q = \{\max(\mathrm{SK}_{lj}), \min(\mathrm{SK}_{lj})\}$. As a result,

$$\begin{aligned}
w_1 &= 2\sum_{\mathrm{SK}_{ij}\notin Q} d_{ij}e_{ij} - \frac{2}{N-2}\sum_{\mathrm{SK}_{ij}\notin Q} d_{ij}\sum_{\mathrm{SK}_{ij}\notin Q} e_{ij}, \\
w_2 &= 2\sum_{\mathrm{SK}_{ij}\notin Q} e_{ij}^2 - \frac{2}{N-2}\left(\sum_{\mathrm{SK}_{ij}\notin Q} e_{ij}\right)^2 + 4\sum_{\mathrm{SK}_{ij}\notin Q} d_{ij}f_{ij} \\
&\quad - \frac{4}{N-2}\sum_{\mathrm{SK}_{ij}\notin Q} d_{ij}\sum_{\mathrm{SK}_{ij}\notin Q} f_{ij}, \\
w_3 &= 6\sum_{\mathrm{SK}_{ij}\notin Q} e_{ij}f_{ij} - \frac{6}{N-2}\sum_{\mathrm{SK}_{ij}\notin Q} e_{ij}\sum_{\mathrm{SK}_{ij}\notin Q} f_{ij}, \\
w_4 &= 4\sum_{\mathrm{SK}_{ij}\notin Q} f_{ij}^2 - \frac{4}{N-2}\left(\sum_{\mathrm{SK}_{ij}\notin Q} f_{ij}\right)^2.
\end{aligned} \tag{36}$$

Table 2: An example.

| | $R_i$ | $\mathrm{RCTE}_{ij}$ | $\mathrm{SK}_{ij}$ | $\mathrm{SK}'_{ij}$ |
|---|---|---|---|---|
| 1 | 319 | 26 | 5.00 | 209.34 |
| 2 | 344 | 61 | 2.59 | 178.78 |
| 3 | 376 | 91 | 2.50 | 172.87 |
| 4 | 311 | 178 | 0.25 | — |
| 5 | 399 | 146 | 1.86 | 146.73 |
| 6 | 325 | 155 | 0.54 | 95.07 |
| 7 | 381 | 16 | 44.77 | 304.73 |
| 8 | 377 | 172 | 1.22 | 116.05 |
| 9 | 319 | 163 | 0.40 | 85.87 |
| 10 | 300 | 12 | $1.12E+08$ | — |
| 11 | 384 | 35 | 8.92 | 241.39 |
| 12 | 381 | 163 | 1.37 | 124.71 |
| 13 | 390 | 97 | 2.68 | 176.87 |
| 14 | 394 | 188 | 1.34 | 116.78 |
| 15 | 364 | 15 | 44.58 | 296.28 |
| 16 | 305 | 17 | 8.35 | 230.23 |
| 17 | 382 | 84 | 2.91 | 182.35 |
| 18 | 359 | 78 | 2.37 | 171.69 |
| 19 | 321 | 71 | 1.25 | 152.30 |
| 20 | 367 | 36 | 7.00 | 225.99 |

In the previous example, after excluding the minimum and maximum slack values, the optimal value of $\xi$ was determined to be 0.11. We compared the results associated with the two settings in Figure 3. Obviously, the second setting achieved better slack diversification because it excluded the minimum and maximum slack values.

## 4. Experimental Results and Discussions

The effectiveness of the biobjective slack-diversifying nonlinear fluctuation smoothing rule was assessed with simulated data. To this end, a memory fabrication factory was simulated with a monthly capacity of up to 32,000 wafers. In the wafer fabrication factory, more than 500 workstations

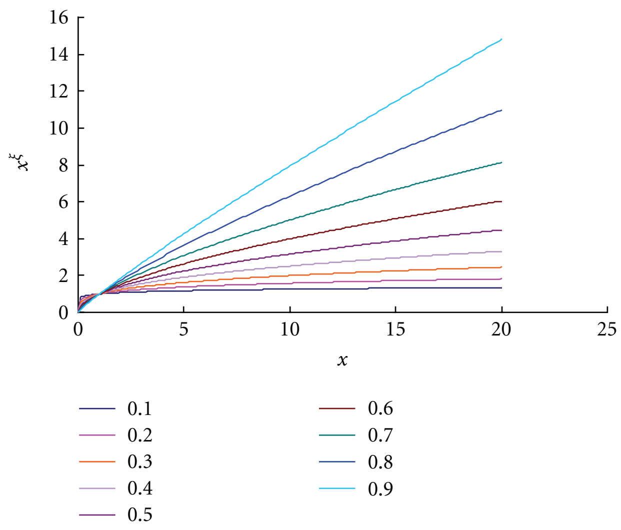
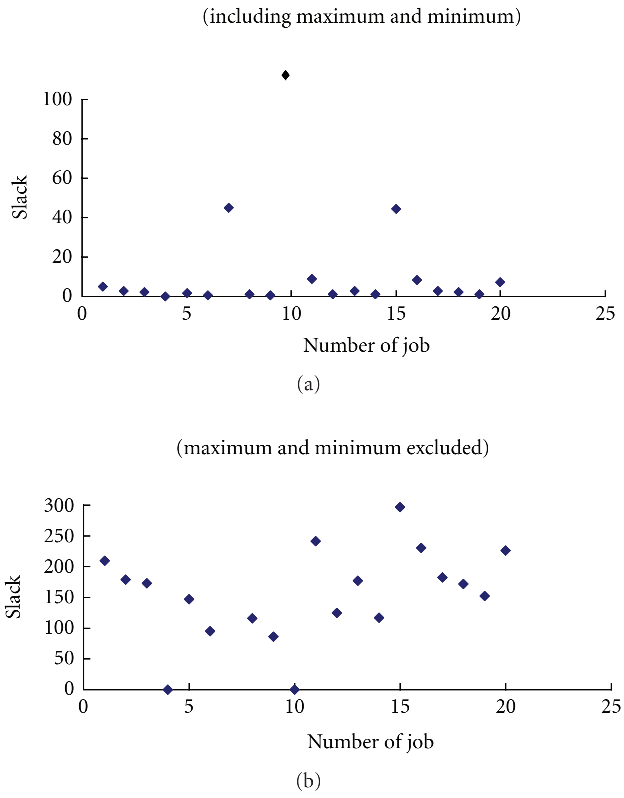


FIGURE 2: The linear approximation of $x^{\xi}$.



(a)

(b)

FIGURE 3: The effects of excluding the maximum and the minimum.

were devoted to single-wafer or batch production using 58 nm~110 nm technologies. The large-scale and the reentrant process flows made production control in the wafer fabrication factory a very tough task. The release policy was uniform; that is, jobs were released into it at a fixed interval, as is common in memory fabrication factories. FIFO was employed to sequence jobs on most of the workstations. The research sought to replace FIFO with better rules that might shorten the average cycle times and quicken deliveries to customers.

Although there were more than 10 products in the wafer fabrication factory, this research only considered the two major products that occupied most of the factory capacity; these were labeled A and B. The simulated jobs were assigned various priorities. Jobs with higher priorities were to be processed first.

Nine existing approaches, FIFO, earliest due date (EDD), shortest remaining processing time (SRPT), CR, FSVCT, FSMCT, 1f-TNFSVCT, 1f-TNFSMCT, and 1f-biNFS, were evaluated for the simulated data. In EDD and CR, the internal due date of a job was determined by changing the cycle time multiplier [19]. Then, from several possible values, the value that gave the best performance was chosen (see Figures 4 and 5). Eleven values of $\xi$ in 1f-TNFSMCT and 1f-TNFSVCT were taken from a list of possible values (0.1, 0.2, . . ., 1) and the $\xi$-value that returned the best schedule was taken as the output of the rule. The value of the factor in 1f-biNFS was determined in a similar way. The average cycle time, cycle time standard deviation of each product, and priority were compared for all approaches, as summarized in Tables 3 and 4.

(1) Table 3 compares the performance levels of these methods with respect to the average cycle time. From the tabulated results, it is obvious that the biobjective slack-diversifying nonlinear fluctuation smoothing rule effectively shortened the average cycle times; for product B with normal priority, it was more than 10% better than FIFO. All the compared approaches were inferior to the biobjective slack-diversifying nonlinear fluctuation smoothing rule in this respect.

(2) At the same time, it can be seen from Table 3 that the cycle time standard deviation was also controlled by applying the biobjective slack-diversifying nonlinear fluctuation smoothing rule. For a job of product A with the greatest time requirement and superhigh priority, the deviation of the cycle time from the average value was only 13 hours. This is remarkable for job dispatching in a wafer fabrication factory and conduces to reliable due date promises.

(3) From Figures 4 and 5, it is obvious that the effects of the cycle time multiplier on EDD and CR were quite different, even though they employed the same method to determine the internal due date.

(4) The biobjective slack-diversifying nonlinear fluctuation smoothing rule was better than the 1f-biNFS, with regard to both the average cycle time and the cycle time standard deviation. The advantages were 9% and 25% on average, respectively, which confirmed the usefulness of factor optimization to tailored rules like 1f-biNFS.

To determine whether the differences between the performance of the biobjective slack-diversifying nonlinear fluctuation smoothing rule and those of the nine existing approaches were significant, the following hypotheses were tested.

Table 3: The performances of various approaches in the average cycle time.

| Avg. cycle time (hrs) | A (normal) | A (hot) | A (super hot) | B (normal) | B (hot) |
|---|---|---|---|---|---|
| FIFO | 1256 | 401 | 320 | 1278 | 457 |
| EDD | 1087 | 346 | 306 | 1433 | 478 |
| SRPT | 966 | 350 | 309 | 1737 | 483 |
| CR | 1143 | 356 | 301 | 1497 | 470 |
| FSMCT | 1401 | 405 | 320 | 1408 | 430 |
| FSVCT | 1046 | 385 | 317 | 1745 | 519 |
| 1f-TNFSMCT | 1353 | 379 | 298 | 1271 | 409 |
| 1f-TNFSVCT | 1443 | 374 | 295 | 1326 | 398 |
| 1f-biNFS | 1351 | 363 | 281 | 1285 | 413 |
| The proposed methodology | 1161 | 310 | 279 | 1145 | 394 |

Table 4: The performances of various approaches in cycle time standard deviation.

| Cycle time std. dev. (hrs) | A (normal) | A (hot) | A (super hot) | B (normal) | B (hot) |
|---|---|---|---|---|---|
| FIFO | 56 | 24 | 23 | 87 | 40 |
| EDD | 130 | 25 | 23 | 50 | 39 |
| SRPT | 246 | 32 | 23 | 106 | 30 |
| CR | 68 | 30 | 19 | 58 | 37 |
| FSMCT | 42 | 44 | 23 | 35 | 28 |
| FSVCT | 319 | 35 | 28 | 222 | 55 |
| 1f-TNFSMCT | 81 | 43 | 22 | 49 | 25 |
| 1f-TNFSVCT | 44 | 28 | 18 | 31 | 21 |
| 1f-biNFS | 65 | 44 | 19 | 41 | 31 |
| The proposed methodology | 63 | 23 | 13 | 40 | 17 |

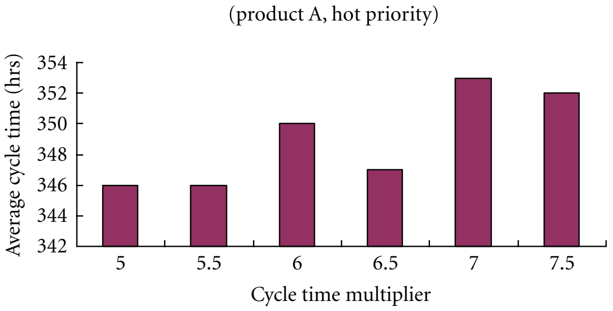


Figure 4: The effects of the cycle time multiplier on EDD.

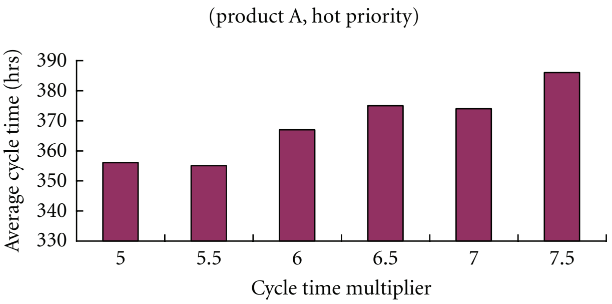


Figure 5: The effects of the cycle time multiplier on CR.

$H_{a0}$: The shortening of the average cycle time of the biobjective slack-diversifying nonlinear fluctuation smoothing rule is the same as that of the compared existing approach.

$H_{a1}$: The shortening of the average cycle time of the biobjective slack-diversifying nonlinear fluctuation smoothing rule is better than that of the compared existing approach.

$H_{b0}$: The reduction in cycle time standard deviation of the biobjective slack-diversifying nonlinear fluctuation smoothing rule is the same as that of the compared existing approach.

$H_{b1}$: The reduction in cycle time standard deviation of the biobjective slack-diversifying nonlinear fluctuation smoothing rule is better than that of the compared existing approach.

Several statistical methods have been developed for testing these hypotheses at a specified significance level $\alpha$. One of the most commonly used methods is the Wilcoxon sign-rank test. The results are summarized in Table 5. The null hypothesis $H_a$ was rejected at $\alpha = 0.025$ or 0.05; the biobjective slack-diversifying nonlinear fluctuation smoothing rule was superior to six existing approaches in reducing the average cycle time. Furthermore, the advantage of the biobjective slack-diversifying nonlinear fluctuation smoothing rule over

Table 5: The results of Wilcoxon sign-rank test.

| | $H_{a0}$ | $H_{b0}$ |
|---|---|---|
| FIFO | $Z = 2.02^{**}$ | $Z = 1.48$ |
| EDD | 1.21 | $2.02^{**}$ |
| SRPT | 0.94 | $2.02^{**}$ |
| CR | $1.75^{*}$ | $2.02^{**}$ |
| FSMCT | $2.02^{**}$ | 0.67 |
| FSVCT | 1.21 | $2.02^{**}$ |
| 1f-TNFSMCT | $2.02^{**}$ | $2.02^{**}$ |
| 1f-TNFSVCT | $2.02^{**}$ | $-0.40$ |
| 4f-biNFS | $2.02^{**}$ | $2.02^{**}$ |

$^{*}P < 0.05$, $^{**}P < 0.025$, $^{***}P < 0.01$.

six existing dispatching rules in reducing cycle time standard deviation was also significant at $\alpha = 0.025$.

## 5. Conclusions and Directions for Future Research

In this paper, we have presented a biobjective slack-diversifying nonlinear fluctuation smoothing rule modified from 1f-biNFS. Our new rule provides superior performance for job dispatching in a wafer fabrication factory. Our new rule maximizes the standard deviation of the slack dynamically; many studies have considered this feature to be conducive to scheduling performance.

A simulation experiment was set up to validate the effectiveness of the biobjective slack-diversifying nonlinear fluctuation smoothing rule.

(1) The biobjective slack-diversifying nonlinear fluctuation smoothing rule incorporates the concept of factor optimization, so as to avoid the drawbacks of existing tailored nonlinear fluctuation smoothing rules. Through self-adjustment and continuous response to the changing conditions in the wafer fabrication factory, the biobjective slack-diversifying nonlinear fluctuation smoothing rule proved itself to be an effective dispatching rule in the simulation experiment.

(2) The effectiveness of the biobjective slack-diversifying nonlinear fluctuation smoothing rule was fully revealed by the overall improvement in the scheduling performance, which was also examined and confirmed by statistical analyses.

(3) The biobjective nature of the biobjective slack-diversifying nonlinear fluctuation smoothing rule was best revealed by the simultaneous improvements in the average cycle time and cycle time standard deviation.

Conversely, there are also disadvantages or limitations associated with the proposed methodology.

(1) The way of diversifying the slack in the proposed methodology is subjective. For the same purpose, there are many other possible ways that can be tried to achieve better performance.

(2) Compared with the existing dispatching rules, the proposed method requires more time to estimate the remaining cycle time and optimizing the rule content.

However, the same concept can be applied to optimize other rules to pursue better scheduling performance. This might be examined in future studies. In addition, to further evaluate the advantages and disadvantages of the proposed methodology, it has to be applied to a full-scale actual semiconductor factory.

## Acknowledgment

This work was supported by National Science Council of Taiwan.

## References

[1] C. N. Wang and C. H. Wang, "A simulated model for cycle time reduction by acquiring optimal lot size in semiconductor manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 34, no. 9-10, pp. 1008–1015, 2007.

[2] H. Lehouche, H. Guéguen, and B. Mendil, "Supervisory control based on closed-loop adaptive control approach of nonlinear systems: application to CSTR process," *Asian Journal of Control*, vol. 14, no. 1, pp. 258–270, 2012.

[3] T. Chen and Y. C. Lin, "A fuzzy-neural fluctuation smoothing rule for scheduling jobs with various priorities in a miconductor manufacturing factory," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 17, no. 3, pp. 397–417, 2009.

[4] T. Chen and Y. C. Wang, "A nonlinear scheduling rule incorporating fuzzy-neural remaining cycle time estimator for scheduling a semiconductor manufacturing factory-a simulation study," *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 110–121, 2009.

[5] T. Chen, "Optimized fuzzy-neuro system for scheduling wafer fabrication," *Journal of Scientific and Industrial Research*, vol. 68, no. 8, pp. 680–685, 2009.

[6] T. Chen, "Dynamic fuzzy-neural fluctuation smoothing rule for jobs scheduling in a wafer fabrication factory," *Proceedings of the Institution of Mechanical Engineers. Part I*, vol. 223, no. 8, pp. 1081–1094, 2009.

[7] T. Chen and Y. C. Wang, "A bi-criteria nonlinear fluctuation smoothing rule incorporating the SOM-FBPN remaining cycle time estimator for scheduling a wafer fab-a simulation study," *International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5–8, pp. 709–721, 2010.

[8] D. A. Koonce and S. C. Tsai, "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule," *Computers and Industrial Engineering*, vol. 38, no. 3, pp. 361–374, 2000.

[9] B. W. Hsieh, C. H. Chen, and S. C. Chang, "Scheduling semiconductor wafer fabrication by using ordinal optimization-based simulation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 599–608, 2001.

[10] H. J. Yoon and W. Shen, "A multiagent-based decision-making system for semiconductor wafer fabrication with hard

temporal constraints," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 83–91, 2008.

[11] K. Sourirajan and R. Uzsoy, "Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication," *Journal of Scheduling*, vol. 10, no. 1, pp. 41–65, 2007.

[12] Y. Harrath, B. Chebel-Morello, and N. Zerhouni, "A genetic algorithm and data mining based meta-heuristic for job shop scheduling problem," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 280–285, October 2002.

[13] S. C. H. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 374–388, 1994.

[14] T. C. Chiang and L. C. Fu, "A simulation study on dispatching rules in semiconductor wafer fabrication facilities with due date-based objectives," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 4660–4665, October 2006.

[15] K. Altendorfer, B. Kabelka, and W. Stöcher, "A new dispatching rule for optimizing machine utilization at a semiconductor test field," in *Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC '07)*, pp. 188–193, June 2007.

[16] H. Zhang, Z. Jiang, and C. Guo, "Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology," *International Journal of Advanced Manufacturing Technology*, vol. 41, no. 1-2, pp. 110–121, 2009.

[17] T. Chen, "Fuzzy-neural-network-based fluctuation smoothing rule for reducing the cycle times of jobs with various priorities in a wafer fabrication plant: a simulation study," *Proceedings of the Institution of Mechanical Engineers, Part B*, vol. 223, no. 8, pp. 1033–1043, 2009.

[18] T. Chen, "A tailored non-linear fluctuation smoothing rule for semiconductor manufacturing factory scheduling," *Proceedings of the Institution of Mechanical Engineers. Part I*, vol. 223, no. 2, pp. 149–160, 2009.

[19] T. Chen, Y. C. Wang, and Y. C. Lin, "A bi-criteria four-factor fluctuation smoothing rule for scheduling jobs in a wafer fabrication factory," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 10, pp. 4289–4303, 2010.

[20] T. Chen, "An optimized tailored nonlinear fluctuation smoothing rule for scheduling a semiconductor manufacturing factory," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 317–325, 2010.

[21] C. Chen, "A self-organized neuro-fuzzy system for air cargo and airline passenger dynamics modeling and forecasting," *International Journal of Fuzzy System Applications*, vol. 2, no. 2, pp. 36–49, 2012.

[22] S. Dehuri and S. Cho, "Learning fuzzy network using sequence bound global particle swarm optimizer," *International Journal of Fuzzy System Applications*, vol. 2, no. 1, pp. 54–70, 2012.

[23] M. Mohammadzaheri, L. Chen, and S. Grainger, "A critical review of the most popular types of neuro control," *Journal of Control*, vol. 14, no. 1, pp. 1–11, 2012.

[24] T. Chen, Y. C. Wang, and H. C. Wu, "A fuzzy-neural approach for remaining cycle time estimation in a semiconductor manufacturing factory-a simulation study," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2125–2139, 2009.

[25] T. Chen, "Job cycle time estimation in a wafer fabrication factory with a bi-directional classifying fuzzy-neural approach," *International Journal of Advanced Manufacturing Technology*, vol. 56, no. 9–12, pp. 1007–1018, 2011.

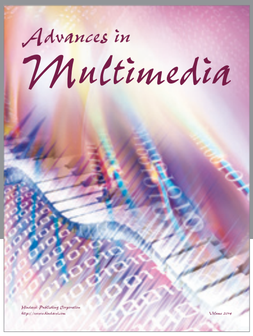
[26] R. Winkler, F. Klawonn, and R. Kruse, "Fuzzy c-means in high dimensional spaces," *International Journal of Fuzzy System Applications*, vol. 1, no. 1, pp. 1–16, 2011.

[27] M. H. Zarandi and Z. S. Razaee, "A fuzzy clustering model for fuzzy data with outliers," *International Journal of Fuzzy System Applications*, vol. 1, no. 2, pp. 29–42, 2011.

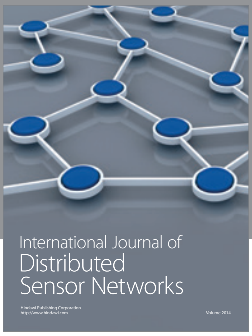
[28] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

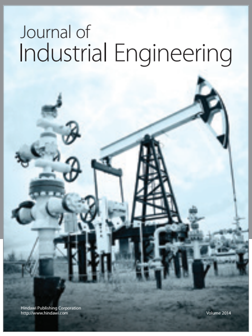
[29] T. Chen, "A fuzzy back propagation network for output time prediction in a wafer fab," *Applied Soft Computing Journal*, vol. 2, no. 3, pp. 211–222, 2003.

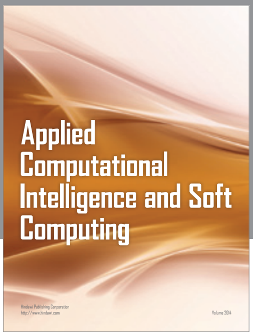
[30] P. C. Pendharkar, "A comparison of gradient ascent, gradient descent and genetic-algorithm- based artificial neural networks for the binary classification problem," *Expert Systems*, vol. 24, no. 2, pp. 65–86, 2007.

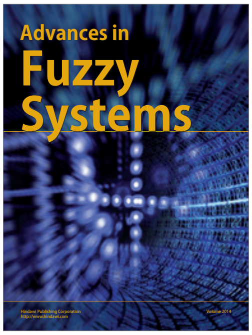

Advances in
Multimedia

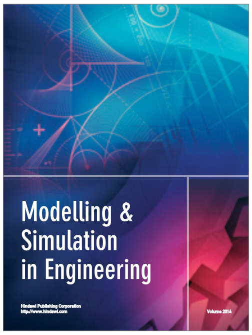
The Scientific
World Journal

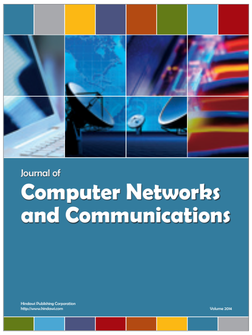

International Journal of
Distributed
Sensor Networks

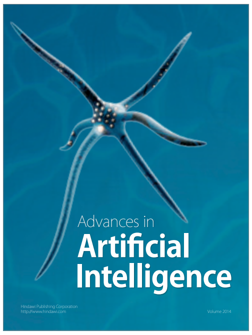

Journal of
Industrial Engineering

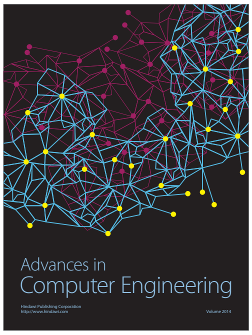

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

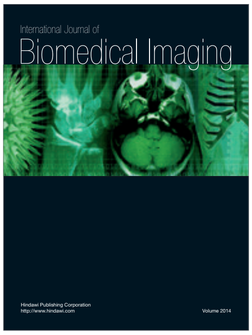

Modelling &
Simulation
in Engineering

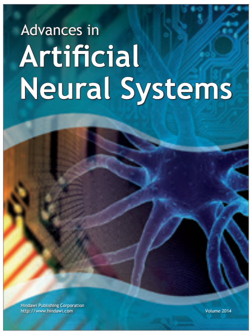

Journal of
Computer Networks
and Communications

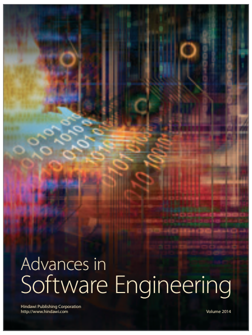

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

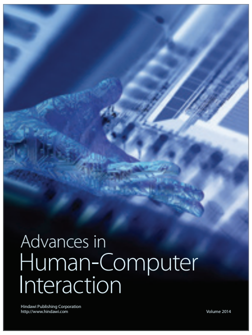
International Journal of
Computer Games
Technology

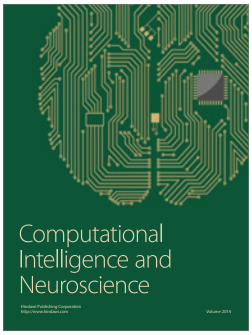

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering