International Journal of Innovative Computing, Information and Control
Volume **9**, Number **6**, June **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 2283–2302

THE OPTIMIZED-RULE-FUSION AND CERTAIN-RULE-FIRST APPROACH FOR MULTI-OBJECTIVE JOB SCHEDULING IN A WAFER FABRICATION FACTORY

Toly Chen

Department of Industrial Engineering and Systems Management
Feng Chia University
No. 100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
tolychen@ms37.hinet.net

Received March 2012; revised July 2012

Abstract. *This paper presents an optimized-rule-fusion and certain-rule-first approach to improve the performance of scheduling jobs in a wafer fabrication factory. As a nonlinear fusion of four traditional dispatching rules, the new rule is aimed at the simultaneous optimization of the average cycle time, cycle time standard deviation, the maximum lateness, and the number of tardy job at the same time, which has rarely been discussed in the past. In addition, we show that there is a contradiction among the four objectives, and establish a certain-rule-first procedure to resolve the contradiction. A more effective fuzzy-neural approach is also applied to estimate the remaining cycle time of a job. Finally, the values of the adjustable parameters in the nonlinear fusion are optimized using response surface method (RSM) instead of being chosen subjectively as in the previous studies. Some theoretical properties of the new rules are also proven. To assess the effectiveness of the proposed methodology, production simulation is also applied in this study. According to the results of a simulation study, the proposed methodology is better than several existing approaches in improving the four objectives at the same time.*
Keywords: Wafer fabrication, Nonlinear fusion, Response surface method, Multi-objective scheduling, Fuzzy, Neural

1. **Introduction.** The production equipment required by a wafer fabrication factory is very expensive, and must be fully utilized. For this purpose, to ensure that the capacity does not substantially exceed the demand is a perquisite. Subsequently, how to plan the use of the existing capacity to shorten the cycle time, and maximize the turnover rate is an important goal. In this regard, scheduling is undoubtedly a very useful tool.

The scheduling of a wafer fabrication factory usually requires the consideration of many viewpoints. To shorten the cycle time is one, and to meet the due date is another. Both views are equally important because to strengthen the relationship with our customers, delivery to the customer as soon as possible and strictly commit to the promise are two requirements, which corresponds to different objectives in job scheduling that need to be optimized at the same time. However, some objectives are contradictory to each other, making their simultaneous optimization a challenging task. This study is dedicated to the investigation of this issue, which also constitutes the motives for the proposed methodology.

First, shortening the cycle time has many implications, corresponding to the so-called regular measures in the scheduling theorem. Common regular measure includes the maximum completion time (makespan) $C_{\max}$, the sum of the completion times ΣC_j, and others. On-time delivery is another area of concern. Missing the deadline not only incurs a fine, but also increases the possibility of losing customers. Typical measures in this field include the maximum lateness $L_{\max}$, the mean lateness $\bar{L}$, the number of tardy job N_T, and others.

Although there has been a consensus for the importance of multi-objective scheduling, the research in this area is still inadequate, especially for large complex production systems. In the literature, most previous studies addressed multi-objective scheduling problems by combining the basic scheduling rules. Each of these rules is optimal for a single target. Loukil et al. [1] mentioned the five ways to deal with multi-objective scheduling:

(1) Simultaneous (or Pareto) approach: Their combination should be formed in such a way that ensures the performances along different dimensions are Pareto optimal. For minimization problems with K objective functions, if the value of objective function f_k can only be decreased by increasing the value of some other objective function f_j, $k \neq j$, j, $k \in \{1, \ldots, M\}$, then all feasible solutions that fulfill this property are called Pareto optimal solutions.
(2) Utility (or compromise) approach: To simplify the finding of the best solution, the linear or nonlinear combination of the objectives can be optimized instead, the so-called compromise solution.
(3) Goal programming (or satisfying) approach: Some objectives are formulated as constraints, for which the satisfaction levels are defined.
(4) Hierarchical approach: Objectives are not optimized at the same time but sequentially.
(5) Interactive approach: A number of steps are required, and the decision maker expresses his/her preferences to the solution proposed at each step to evolve it to the most acceptable one.

Grimme and Lepping [2] considered bi-objective single-machine scheduling problems. They mentioned that there are two ways to solve multi-objective scheduling problems – problem-specific approaches [3] and black-box optimizers like randomized search or evolutionary algorithms. van Wassenhoven and Gelder [4] proposed an efficient algorithm for the $1||\Sigma C_j$, $L_{\max}$ problem, where ΣC_j and $L_{\max}$ means the total completion time and maximum lateness, respectively. In order to optimize the total completion time and makespan on m identical machines, i.e., the $P_m||\Sigma C_j$, $C_{\max}$ problem, Stein and Wein [5] proposed a general algorithmic framework. In these methods, some operations such as truncation and composition are applied to merge schedules that are optimal for different goals, assuming that after the merger the new schedule is still satisfactory for the goals. Cochran et al. [6] proposed a multi-population genetic algorithm to solve two-objective (makespan and total weighted tardiness) scheduling problems for parallel machines. Similar to a compromise solution, the multiplication of the relative measure of each objective was optimized. Loukil et al. [1] proposed a multi-objective simulated annealing to solve multiple-objective scheduling problems for one machine, parallel machines, and permutation flow shops. Most cases contained at most two objectives, and the only case with three objectives was on a single machine.

On the other hand, a detailed review of the application of evolutionary algorithms in multi-objective scheduling can be found in Deb [7] and Coello et al. [8]. Most studies used genetic algorithms (GAs) to evolve some good schedules to approach the optimal schedule.

Multi-objective scheduling in a wafer fabrication factory is difficult because of several reasons:

(1) A wafer fabrication factory is a very complex manufacturing system featured by changing demand, a variety of product types and priorities, un-balanced capacity, job reentry into machines, alternative machines with unequal capacity, sequence dependent setup times, and shifting bottlenecks, so that job dispatching in a wafer fabrication factory becomes a very difficult task.
(2) For such a complex production system, it is even difficult to find a heuristic to optimize a single objective [3,9,10].
(3) A naive aggregation of single-objective heuristic does not necessarily yield feasible non-dominated solutions [2].
(4) Considering the weighted sum of the objectives often leads to unsatisfactory results.

A few studies applied the response surface method (RSM) and the desirability function to handle multiple-factor and multiple-objective optimization in scheduling [11]. However, the commonly used second-order multiple-factor regression may not be accurate enough. The desirability function is a very subjective approach. In the final, most dispatching rules are focused on a single performance measure. In fact, minimizing any performance measure in such a complex job shop is a strongly NP-hard problem. However, optimizing multiple performance measures at the same time is still being pursued. Taking into account two performance measures (average cycle time and cycle time variation) at the same time, Chen [3] proposed a bi-criteria nonlinear fluctuation smoothing rule that also has an adjustable factor (1f-biNFS). To increase the flexibility of customization, Chen et al. [12] extended the above rules, and proposed the bi-criteria fluctuation smoothing rule with four adjustable factors (4f-biNFS). However, the adjustment factors in these rules are static. In other words, they will not change over time. Chen [9] therefore designed a mechanism to dynamically adjust the values of the factors in Chen and Wang's bi-criteria nonlinear fluctuation smoothing rule (dynamic 1f-biNFS). However, the adjustment of the factors is based on a pre-defined rule. This process is too subjective, and does not also take into account the status of the wafer fabrication factory. In addition, these rules have not been optimized, so there is considerable room for improvement. A few studies applied the response surface method (RSM) and the desirability function to handle multiple-factor and multiple-objective optimization in scheduling [11]. However, the commonly used second-order multiple-factor regression may not be accurate enough. The desirability function is a very subjective approach. In the final, most dispatching rules are focused on a single performance measure. In fact, minimizing any performance measure in such a complex job shop is a strongly NP-hard problem. However, optimizing multiple performance measures at the same time is still being pursued. Taking into account two performance measures (average cycle time and cycle time variation) at the same time, Chen [3] proposed a bi-criteria nonlinear fluctuation smoothing rule that also has an adjustable factor (1f-biNFS). To increase the flexibility of customization, Chen et al. [12] extended the above rules, and proposed the bi-criteria fluctuation smoothing rule with four adjustable factors (4f-biNFS). However, the adjustment factors in these rules are static. In other words, they will not change over time. Chen [9] therefore designed a mechanism to dynamically adjust the values of the factors in Chen and Wang's bi-criteria nonlinear fluctuation smoothing rule (dynamic 1f-biNFS). However, the adjustment of the factors is based on a pre-defined rule. This process is too subjective, and does not also take into account the status of the wafer fabrication factory. In addition, these rules have not been optimized, so there is considerable room for improvement.

A rule-fusion and certain-rule-first procedure is established in this study for multi-objective job scheduling in a wafer fabrication factory. The unique features of the proposed methodology include:

(1) Four performance measures – the average cycle time, cycle time standard deviation, the maximum lateness, and the number of tardy jobs, are optimized at the same time. As far as we know, the existing dispatching rules in this field were not designed for this purpose. Most of them considered at most two objectives, or only evaluated the performance of the rules along multiple dimensions for simple systems, let alone for such a complex manufacturing system.
(2) We propose a more effective fuzzy-neural approach to estimate the remaining cycle time of a job. The fuzzy-neural approach is based on the fuzzy c-means and back propagation network (FCM-BPN) approach [13,14]. According to Chen and Wang [10], with more accurate remaining cycle time estimation, the scheduling performance of a fluctuation smoothing rule can be significantly improved. In the original study, Chen and Wang used a gradient search algorithm for training the BPN, which is time-consuming and not very accurate. In this study, we use the Levenberg-Marquardt algorithm to achieve the same purpose, which is more efficient than that in Chen and Wang's study, and can produce more accurate forecasts.
(3) The new rule is formed by fusing four traditional dispatching rules – the fluctuation smoothing rule for mean cycle time (FSMCT), the fluctuation smoothing rule for cycle time variation (FSVCT), the earliest due date (EDD) rule, and the critical ratio (CR) rule, in a nonlinear way. The combination of so many rules has not been attempted in the literature, because there is in fact a contradiction between them. Nevertheless, such a treatment has two advantages [9,10]. First, the effects of the parameters in the rule can be balanced better. Second, the new rule is more responsive to the changes in the parameters.
(4) We also show that in theory there is a contradiction between {FSMCT, FSVCT} and {EDD, CR}. To resolve the contradiction, a certain-rule-first procedure is established. The philosophy behind the certain-rule-first procedure is to retain the ranks of jobs with the highest or lowest priorities in the original rules.
(5) The content of the new dispatching rule can be tailored for a specific wafer fabrication factory with five adjustable factors. In the similar methods that have recently been proposed, the values of the adjustable parameters were determined in a subjective way. For example, Chen et al. [12] proposed three models, the linear, nonlinear, and logarithmic models, to generate these values. Only nine combinations were chosen subjectively, which could not guarantee the optimality of the scheduling performance. In this study, we determine the optimal values of the parameters using four-dimensional RSM, so as to optimize the scheduling performance. The new dispatching rule can be localized. It can even be tailored to each machine in the wafer fabrication factory. In the literature (e.g., [11]), most heuristics can only be tailored to two machines in a wafer fabrication factory – one for bottlenecks and the other one for non-bottlenecks.
(6) We also discuss the properties of the new dispatching rule from a theoretical point of view.

The differences between the proposed methodology and the previous methods are summarized in Table 1. To assess the effectiveness of the proposed methodology, production simulation is also applied in this study. The rest of this paper is organized as follows. Section 2 is divided into two parts. In the first part, a more effective fuzzy-neural approach is applied to estimate the remaining cycle time of a job. Subsequently, the rule-fusion and certain-rule-first procedure for multi-objective job scheduling in a wafer fabrication factory is detailed. To assess the effectiveness of the proposed methodology, a simulated case study is described in Section 3. According to the results of the analyses, some discussion points are made. Finally, concluding remarks are made in Section 4.

TABLE 1. The differences between the proposed methodology and the previous methods

Rule	Number of objectives	Objectives	Number of adjustable parameters	Optimized?	How to derive the rule?
NFSMCT	1	average cycle time	1	No	• generalizing FSMCT
1f-TNFSVCT	1	cycle time standard variation	1	No	• generalizing FSVCT • adding adjustable parameters
1f-TNFSMCT	1	average cycle time	1	No	• generalizing FSMCT • adding adjustable parameters
2f-TNFSVCT	1	cycle time standard deviation	2	No	• generalizing FSVCT • adding adjustable parameters
4f-biNFS	2	average cycle time, cycle time standard deviation	4	No	• fusing FSVCT and FSMCT • adding adjustable parameters
The proposed methodology	4	average cycle time, cycle time standard deviation, the maximum lateness, the number of tardy jobs	5	Yes	• fusing FSMCT, FSVCT, EDD, and CR • adding adjustable parameters • certain-rule-first policy • RSM

2. **Methodology.** The variables and parameters that will be used in the rule-fusion and certain-rule-first procedure are defined in the following:

(1) R_j: the release time of job j; $j = 1 \sim n$.
(2) BQ_j: the total queue length before bottlenecks at R_j.
(3) CR_{ju}: the critical ratio of job j at step u.
(4) CT_j: the cycle time of job j.
(5) CTE_j: the estimated cycle time of job j.
(6) D_j: the average delay of the three most recently completed jobs at R_j.
(7) DD_j: the due date of job j.
(8) FQ_j: the total queue length in the whole factory at R_j.
(9) Q_j: the queue length on the processing route of job j at R_j.
(10) $RCTE_{ju}$: the estimated remaining cycle time of job j from step u.
(11) RPT_{ju}: the remaining processing time of job j from step u.
(12) SCT_{ju}: the step cycle time of job j until step u.
(13) SK_{ju}: the slack of job j at step u.
(14) U_j: the average factory utilization before job j is released. If the utilization of the factory is reported on a daily basis, then U_j is the utilization of the day before job j is released.
(15) WIP_j: the factory work-in-progress (WIP) at R_j.
(16) λ: mean release rate.
(17) x_p: inputs to the three-layer BPN, $p = 1 \sim 6$.

(18) h_l: the output from hidden-layer node l, $l = 1 \sim L$.
(19) w_l^o: the connection weight between hidden-layer node l and the output node.
(20) w_{pl}^h: the connection weight between input node p and hidden-layer node l, $p = 1 \sim 6$; $l = 1 \sim L$.
(21) θ_l^h: the threshold on hidden-layer node l.
(22) θ^o: the threshold on the output node.

The rule-fusion and certain-rule-first procedure includes the following eight steps:

(1) Normalize the collected data.
(2) Use FCM to classify jobs. The required inputs for this step are job attributes. To determine the optimal number of categories, we use the S test. The output of this step is the category of each job.
(3) Use the BPN approach to estimate the cycle time of each job. Jobs of different categories will be sent to different three-layer BPNs. The inputs to the three-layer BPN include the attributes of a job, while the output is the estimated cycle time of the job.
(4) Derive the remaining cycle time of each job from the estimated cycle time.
(5) Form the new rule by fusing four traditional rules – EDD, CR, FSMCT, and FSVCT in a nonlinear way.
(6) Incorporate the estimated remaining cycle time into the new rule.
(7) Propose the certain-rule-first policy.
(8) Determine the optimal values of the adjustable parameters using RSM.

2.1. **The FCM approach.** The remaining cycle time of a job being produced in a wafer fabrication factory is the time still needed to complete the job. If the job is just 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 attribute (or performance measure) for the WIP in the wafer fabrication factory. Why do we need to estimate the remaining cycle time for each job? This is because the remaining cycle time is an important input for the scheduling rule. Past studies (e.g., [3]) have shown that the accuracy of the remaining cycle time forecasting can be improved by job classification. Soft computing methods (e.g., [15]) have received much attention in this field.

In the intelligent approach, jobs are classified into K categories using FCM. Why is a fuzzy clustering method such as FCM applied? If a crisp clustering method is applied, then it is possible that some clusters will have very few examples. In contrast, an example belongs to multiple clusters to different degrees in a fuzzy clustering method, which provides a solution to this problem.

First, in order to facilitate the subsequent calculations and problem solving, all raw data are normalized into [0.1 0.9] [16]:

$$N(x) = 0.1 + \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (0.9 - 0.1) \tag{1}$$

where $N(x)$ is the normalized value of x; $x_{\min}$ and $x_{\max}$ are the minimum and maximum of x, respectively. In this way, it is possible that the future value of x is greater than $x_{\max}$ or less than $x_{\min}$. The formula can be written as

$$x = \frac{N(x) - 0.1}{0.9 - 0.1} \cdot (x_{\max} - x_{\min}) + x_{\min} \tag{2}$$

if the un-normalized value is to be obtained instead. Then, we place the (normalized) attributes of a job in vector $\boldsymbol{x}_j = [x_{j1}, x_{j2}, x_{j3}, x_{j4}, x_{j5}, x_{j6}] = [N(U_j), N(Q_j), N(BQ_j), N(FQ_j), N(WIP_j), N(D_j)]$. Although the job size is an important factor, we consider only full-size jobs and therefore the job size becomes a constant.

FCM classifies jobs by minimizing the following objective function:

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

where K is the required number of categories; n is the number of jobs; $\mu_{j(k)}$ indicates the membership that job j belongs to category k; $e_{j(k)}$ measures the distance from job j to the centroid of category k; $m \in [1, \infty)$ is a parameter to adjust the fuzziness, and is usually set to 2. The procedure of FCM is as follows:

(1) Produce a preliminary clustering result.

(2) (Iterations) Calculate the centroid of each category as

$$\bar{x}_{(k)} = \{\bar{x}_{(k)p}\}; \quad p = 1 \sim 6 \tag{4}$$

$$\bar{x}_{(k)p} = \sum_{j=1}^{n} \mu_{j(k)}^{m} x_{jp} \Big/ \sum_{j=1}^{n} \mu_{j(k)}^{m} \tag{5}$$

$$\mu_{j(k)} = 1 / \sum_{q=1}^{K} (e_{j(k)}/e_{j(q)})^{2/(m-1)} \tag{6}$$

$$e_{j(k)} = \sqrt{\sum_{all\ p} (x_{jp} - \bar{x}_{(k)p})^2} \tag{7}$$

where $\bar{x}_{(k)}$ is the centroid of category k. $\mu_{j(k)}^{(t)}$ is the membership that job i belongs to category k after the t-th iteration.

(3) Re-measure the distance from each job to the centroid of each category, and then recalculate the corresponding membership.

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

$$\max_{k} \max_{j} \left| \mu_{j(k)}^{(t)} - \mu_{j(k)}^{(t-1)} \right| < d \tag{8}$$

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

Finally, the separate distance test (S test) proposed by Xie and Beni [17] can be applied to determine the optimal number of categories K:

$$\text{Min}\, S \tag{9}$$

subject to

$$J_m = \sum_{k=1}^{K} \sum_{j=1}^{n} \mu_{j(k)}^{m} e_{j(k)}^{2} \tag{10}$$

$$e_{\min}^{2} = \min_{k1 \neq k2} \left(\sum_{all\ p} (\bar{x}_{(k1)p} - \bar{x}_{(k2)p})^2 \right) \tag{11}$$

$$S = \frac{J_m}{n \times e_{\min}^{2}} \tag{12}$$

$$K \in Z^{+} \tag{13}$$

The K value minimizing S determines the optimal number of categories [18,19].

2.2. **The BPN approach.** After clustering, a portion of the jobs in each category is input as the 'training examples' to the three-layer BPN to determine the parameter values. The configuration of the three-layer BPN is set up as follows. First, inputs are the six parameters associated with the j-th example/job including U_j, Q_j, BQ_j, FQ_j, WIP_j, and D_j. These parameters have to be normalized before feeding into the three-layer BPN. Subsequently, there is only a single hidden layer with neurons that are twice that in the input layer. Finally, the output from the three-layer BPN is the (normalized) estimated cycle time (CTE_j) of the example. The activation function used in each layer is Sigmoid function,

$$f(x) = 1/(1 + e^{-x}) \tag{14}$$

The procedure for determining the parameter values is now described. Two phases are involved at the training stage. At first, in the forward phase, inputs are multiplied with weights, summated, and transferred to the hidden layer. Then activated signals are outputted from the hidden layer as

$$h_l = \frac{1}{1 + e^{-n_l^h}} \tag{15}$$

where

$$n_l^h = I_l^h - \theta_l^h \tag{16}$$

$$I_l^h = \sum_{p=1}^{6} w_{pl}^h \cdot x_{jp} \tag{17}$$

h_l's are also transferred to the output layer with the same procedure. Finally, the output of the BPN is generated as

$$o_j = \frac{1}{1 + e^{-n^o}} \tag{18}$$

where

$$n^o = I^o - \theta^o \tag{19}$$

$$I^o = \sum_{l=1}^{L} w_l^o \cdot h_l \tag{20}$$

Subsequently in the backward phase, some algorithms are applicable for training a BPN, such as the gradient descent algorithms, the conjugate gradient algorithms, the Levenberg-Marquardt algorithm, and others. In this study, the Levenberg-Marquardt algorithm is applied. Finally, the three-layer BPN can be applied to estimate the cycle time of a job, and then the remaining cycle time of the job can be derived as

$$RCTE_{ju} = CTE_j - SCT_{ju} \tag{21}$$

When a new job is released into the factory, the six parameters associated with the new job are recorded. Then the new job is classified into a category, so that the three-layer BPN of the category and Equation (21) can be applied to estimate the cycle time and remaining cycle time of the new job, respectively.

2.3. **The four-objective rule.** In traditional fluctuation smoothing (FS) rules there are two different formulation methods, depending on the scheduling purpose [20]. One method is aimed at minimizing the mean cycle time with FSMCT:

$$SK_{ju}(FSMCT) = j/\lambda - RCTE_{ju} \tag{22}$$

The other method is aimed at minimizing the variance of cycle time with FSVCT:

$$SK_{ju}(FSVCT) = R_j - RCTE_{ju} \tag{23}$$

Jobs with the smallest slack values will be given higher priority. These two rules and their variants have been proven to be very effective in shortening the cycle time in wafer fabrication factories [3,9,10,20].

On the other hand, the EDD rule is theoretically the optimal rule to minimize the maximum lateness for simple production systems [21]. It is also the most commonly used dispatching rule by make-to-order (MTO) wafer fabrication factories to achieve on-time delivery. EDD gives higher priorities to jobs with the earliest due dates, so we can define the slack value as:

$$SK_{ju}(EDD) = DD_j \tag{24}$$

If the due date is determined in an internal way, then Equation (24) can be re-written as

$$SK_{ju}(EDD) = R_j + CTE_j + \kappa \tag{25}$$

where κ is a constant allowance. Substituting Equation (21) into Equation (25), we get

$$SK_{ju}(EDD) = R_j + RCTE_{ju} + SCT_{ju} + \kappa \tag{26}$$

Another policy that is considered to be conducive to the minimization of the number of tardy job is the CR policy [22], in which the critical ratio of a job is defined as

$$CR_{ju} = (t - DD_j)/RPT_{ju} \tag{27}$$

So we can define the slack of a job as

$$\begin{aligned} SK_{ju}(CR) &= \frac{1}{CR_{ju}} = \frac{RPT_{ju}}{t - DD_j} = \frac{RPT_{ju}}{t - R_j - CTE_j - \kappa} \\ &= \frac{RPT_{ju}}{t - R_j - RCTE_{ju} - SCT_{ju} - \kappa} = \frac{RPT_{ju}}{\kappa' - R_j - RCTE_{ju} - SCT_{ju}} \end{aligned} \tag{28}$$

where $\kappa' = t - \kappa$ is a constant. If $RCTE_{ju}$ is large, then $SK_{ju}(EDD)$ and $SK_{ju}(CR)$ are high, but $SK_{ju}(FSMCT)$ and $SK_{ju}(FSVCT)$ are low, which means that the four objectives are contradictory in nature.

Theorem 2.1. *Using simple addition and subtraction operators to merge these rules will result in the loss of information.*

Proof: For example, if we use the addition operator to merge FSMCT and EDD:

$$\begin{aligned} SK_{ju}(FSMCT) + SK_{ju}(EDD) &= \frac{j}{\lambda} - RCTE_{ju} + R_j + RCTE_{ju} + SCT_{ju} + \kappa \\ &= \frac{j}{\lambda} + R_j + SCT_{ju} + \kappa \end{aligned} \tag{29}$$

The information contained in $RCTE_{ju}$ is lost. Similarly, the same problem occurs if we combine FSMCT and FSVCT with the subtraction operator:

$$SK_{ju}(FSMCT) - SK_{ju}(FSVCT) = \frac{j}{\lambda} - RCTE_{ju} - R_j + RCTE_{ju} = \frac{j}{\lambda} - R_j \tag{30}$$

Theorem 2.1 implies that the multiplication or division operators may be more suitable for combining these rules.

To solve these problems, the following steps are followed:

(1) Normalize all terms for the four rules. Its purpose is to balance the effects of the parameters in the rules [3].
(2) Calculate SK_{ju} using the multiplication and division operators instead. According to Chen [3], this treatment magnifies the difference in the slack value, which seems to be a good way of improving the scheduling performance.
(3) Derive a general expression of the four types of SK_{ju}'s.

First of all, the components in Equations (22), (23), (26), and (28) are normalized:

$$N(j/\lambda) = \frac{j/\lambda - 1/\lambda}{n/\lambda - 1/\lambda} = \frac{j-1}{n-1} \tag{31}$$

$$N(RCTE_{ju}) = \frac{RCTE_{ju} - \min_j RCTE_{ju}}{\max_j RCTE_{ju} - \min_j RCTE_{ju}} \tag{32}$$

$$N(R_j) = \frac{R_j - \min_j R_j}{\max_j R_j - \min_j R_j} \tag{33}$$

$$N(SCT_{ju}) = \frac{SCT_{ju} - \min_j SCT_{ju}}{\max_j SCT_{ju} - \min_j SCT_{ju}} \tag{34}$$

$$N(RPT_{ju}) = \frac{RPT_{ju} - \min_j RPT_{ju}}{\max_j RPT_{ju} - \min_j RPT_{ju}} \tag{35}$$

where $N()$ is the normalization function. Subsequently, multiplication and division are used to replace the original addition and subtraction operators, respectively. After replacement, Equations (22) and (23) become

$$SK_{ju}(FSMCT) = \frac{j-1}{n-1} \Big/ \frac{RCTE_{ju} - \min_j RCTE_{ju}}{\max_j RCTE_{ju} - \min_j RCTE_{ju}} \tag{36}$$

$$SK_{ju}(FSVCT) = \frac{R_j - \min_j R_j}{\max_j R_j - \min_j R_j} \Big/ \frac{RCTE_{ju} - \min_j RCTE_{ju}}{\max_j RCTE_{ju} - \min_j RCTE_{ju}} \tag{37}$$

while Equations (26) and (28) change to

$$SK_{ju}(EDD) = \frac{R_j - \min_j R_j}{\max_j R_j - \min_j R_j} \cdot \frac{RCTE_{ju} - \min_j RCTE_{ju}}{\max_j RCTE_{ju} - \min_j RCTE_{ju}} \cdot \frac{SCT_{ju} - \min_j SCT_{ju}}{\max_j SCT_{ju} - \min_j SCT_{ju}} \tag{38}$$

$$SK_{ju}(CR) = \frac{RPT_{ju} - \min_j RPT_{ju}}{\max_j RPT_{ju} - \min_j RPT_{ju}} \Big/ \left(\frac{R_j - \min_j R_j}{\max_j R_j - \min_j R_j} \cdot \frac{RCTE_{ju} - \min_j RCTE_{ju}}{\max_j RCTE_{ju} - \min_j RCTE_{ju}} \cdot \frac{SCT_{ju} - \min_j SCT_{ju}}{\max_j SCT_{ju} - \min_j SCT_{ju}} \right) \tag{39}$$

A general expression of the four equations can be derived as follows.

Definition 2.1. *(the four-objective dispatching rule). The slack value of job j at step u according to the four-objective dispatching rule is defined as*

$$SK_{ju}(new\ rule) = \left(\frac{j-1}{n-1}\right)^{\alpha} \cdot \left(\frac{RPT_{ju} - \min_j RPT_{ju}}{\max_j RPT_{ju} - \min_j RPT_{ju}}\right)^{\beta} \cdot \left(\frac{R_j - \min_j R_j}{\max_j R_j - \min_j R_j}\right)^{\gamma} \cdot \left(\frac{RCTE_{ju} - \min_j RCTE_{ju}}{\max_j RCTE_{ju} - \min_j RCTE_{ju}}\right)^{\eta} \cdot \left(\frac{SCT_{ju} - \min_j SCT_{ju}}{\max_j SCT_{ju} - \min_j SCT_{ju}}\right)^{\vartheta} \quad (40)$$

where α, β, γ, and η are positive real numbers satisfying the following constraints:

$$\text{If } \alpha = 1 \text{ then } \beta, \gamma, \vartheta = 0; \quad \eta = -1, \text{ and vice versa} \quad (41)$$

$$\text{If } \beta = 1 \text{ then } \alpha = 0; \quad \gamma, \eta, \vartheta = -1, \text{ and vice versa} \quad (42)$$

$$\text{If } \eta = 1 \text{ then } \alpha, \beta = 0; \quad \gamma, \vartheta = 1, \text{ and vice versa} \quad (43)$$

There are many possible models to form the combinations of α, β, γ, η, and ϑ. For example,

$$\text{(Linear model) } \alpha = 1 - 2\beta - \gamma;\ \gamma = \vartheta = \eta + \alpha \quad (44)$$

$$\text{(Nonlinear model) } \alpha = (1 - 2\beta - \gamma)^u, \quad u \in Z^+; \gamma = \vartheta = (\eta + \alpha)^v, \quad v = 1, 3, 5, \ldots \quad (45)$$

$$\text{(Logarithmic model 1) } \alpha = \ln(2 - 2\beta - \gamma)/\ln 2; \quad \gamma = \vartheta = \ln(1.5\eta + \alpha + 2.5)/\ln 2 - 1 \quad (46)$$

Equation (41) requires the values of α and β to be within [0 1]. *With any model, the proposed methodology tries various combinations of α, β, γ, η, and ϑ to optimize the scheduling performance in the target wafer fabrication factory. In this way, the new rule becomes tailored to the specific wafer fabrication factory. In addition, the values of α, β, γ, η, and ϑ can be dynamically adjusted to reflect the changes in the production conditions of the wafer fabrication factory. Clearly, EDD, FSMCT, FSVCT, and CR are special cases of the new rule:*

EDD: $(\alpha, \beta, \gamma, \eta, \vartheta) = (0, 0, 1, 1, 1)$
FSMCT: $(\alpha, \beta, \gamma, \eta, \vartheta) = (1, 0, 0, -1, 0)$
FSVCT: $(\alpha, \beta, \gamma, \eta, \vartheta) = (0, 0, 1, -1, 0)$
CR: $(\alpha, \beta, \gamma, \eta, \vartheta) = (0, 1, -1, -1, -1)$

Theorem 2.2. *The four-objective nonlinear fluctuation smoothing rule is more responsive than the four original rules if $RCTE_{ju}$ is large, which is a common phenomenon in a wafer fabrication factory.*

Proof: Let us compare the four-objective nonlinear fluctuation smoothing rule and FSMCT first. For a fair comparison, the parameters β, γ, and ϑ are set to 0, because the corresponding variables are not considered in FSMCT.

When $\frac{j}{\lambda}$ increases by 1%, in FSMCT SK_{ju} is changed by

$$\left|\frac{(1+1\%)\frac{j}{\lambda} - RCTE_{ju}}{\frac{j}{\lambda} - RCTE_{ju}} - 1\right| \cdot 100\% = \left|\frac{\frac{j}{\lambda}}{\frac{j}{\lambda} - RCTE_{ju}}\right| \% = \left|\frac{j}{j - \lambda \cdot RCTE_{ju}}\right| \% \quad (47)$$

If RCTE_{ju} is large, such as greater than $\frac{j}{\lambda}$, then Equation (47) becomes

$$\left|\frac{j}{j - \lambda \cdot RCTE_{ju}}\right| \% = \frac{j}{\lambda \cdot RCTE_{ju} - j} \% \quad (48)$$

Conversely, in the four-objective nonlinear fluctuation smoothing rule, SK_{ju} will be changed by

$$\left|\frac{\begin{array}{c}(\frac{(1+1\%)\frac{j}{\lambda}-\frac{1}{\lambda}}{\frac{n}{\lambda}-\frac{1}{\lambda}})^{\alpha}\cdot(\frac{RPT_{ju}-\min_j(RPT_{ju})}{\max_j(RPT_{ju})-\min_j(RPT_{ju})})^{0}\cdot(\frac{R_j-\min_j(R_j)}{\max_j(R_j)-\min_j(R_j)})^{0}\\ \cdot(\frac{RCTE_{ju}-\min_j(RCTE_{ju})}{\max_j(RCTE_{ju})-\min_j(RCTE_{ju})})^{\eta}\cdot(\frac{SCT_{ju}-\min_j(SCT_{ju})}{\max_j(SCT_{ju})-\min_j(SCT_{ju})})^{0}\end{array}}{\begin{array}{c}(\frac{\frac{j}{\lambda}-\frac{1}{\lambda}}{\frac{n}{\lambda}-\frac{1}{\lambda}})^{\alpha}\cdot(\frac{RPT_{ju}-\min_j(RPT_{ju})}{\max_j(RPT_{ju})-\min_j(RPT_{ju})})^{0}\cdot(\frac{R_j-\min_j(R_j)}{\max_j(R_j)-\min_j(R_j)})^{0}\\ \cdot(\frac{RCTE_{ju}-\min_j(RCTE_{ju})}{\max_j(RCTE_{ju})-\min_j(RCTE_{ju})})^{\eta}\cdot(\frac{SCT_{ju}-\min_j(SCT_{ju})}{\max_j(SCT_{ju})-\min_j(SCT_{ju})})^{0}\end{array}}-1\right|\cdot 100\%$$

$$=\left|(\frac{0.01j}{j-1})^{\alpha}\right|\cdot 100\%=(\frac{0.01j}{j-1})^{\alpha}\cdot 100\% \quad (49)$$

because $j \geq 1$. In addition, since $0 \leq \alpha \leq 1$, $1/\alpha \geq 1$ and $100^{\frac{1}{\alpha}} \geq 100^1 = 100$. Therefore, Equation (49) becomes

$$\left(\frac{0.01j}{j-1}\right)^{\alpha}\cdot 100\%=\left(\frac{0.01j}{j-1}\right)^{\alpha}\cdot\left(100^{\frac{1}{\alpha}}\right)^{\alpha}\%\geq\left(\frac{0.01j}{j-1}\right)^{\alpha}\cdot(100)^{\alpha}\%=\left(\frac{j}{j-1}\right)^{\alpha}\% \quad (50)$$

If $RCTE_{ju}$ is large, such as greater than $\frac{(j-1)^{\alpha}+j^{\alpha}}{\lambda j^{\alpha-1}}$, then

$$RCTE_{ju} \geq \frac{(j-1)^{\alpha}+j^{\alpha}}{\lambda j^{\alpha-1}} \quad (51)$$

$$\frac{j}{\lambda RCTE_{ju}-j} \leq (\frac{j}{j-1})^{\alpha} \quad (52)$$

which finishes the proof. The comparison between the four-objective nonlinear fluctuation smoothing rule and the other rules can be done in similar ways.

2.4. **The certain-rule-first procedure.** The contradiction in the four original rules leads to distinct sequencing results by these rules. The problem therefore arises: which rule should we follow in determining the sequence of jobs? In the proposed methodology, we use the mix of them, as detailed in the following certain-rule-first procedure.

Definition 2.2. (the certain rule). *Assume the rank of job j according to FSMCT, FSVCT, EDD, and CR are indicated with $r_j(1)$, $r_j(2)$, $r_j(3)$, and $r_j(4)$, respectively. Obviously, $1 \leq r_j(p) \leq n$; $p = 1 \sim 4$. We say that rule p is the certain rule for job j if*

$$\frac{|r_j(p)-\frac{n}{2}|}{\max_{q\neq p}|r_j(q)-\frac{n}{2}|} > \Theta \quad (53)$$

where Θ is a positive-valued, predefined threshold. If none of the four rules satisfies Equation (53), then the four-objective rule becomes the certain rule for this job. It is also easy to prove that among the four rules and the four-objective rule, only one will be certain if $\Theta > 1$. The philosophy behind the certain-rule-first procedure is to retain the ranks of jobs with the highest or lowest priorities in the original rules. Jobs are also sequenced according to their certain rules.

2.5. **Optimizing the parameters in the four-objective rule.** In most similar rules proposed in the literature, the adjustable parameters are determined in a subjective way. Only a few combinations of them are considered, from which the one giving the best results is chosen. Such a way cannot guarantee the optimality of the scheduling performances. To solve this problem, we can model the relationship between the scheduling performances and the adjustable parameters using RSM. Then, from the response surface, the minimum giving the optimal scheduling performances is chosen.

In this optimization problem, we have five inputs and four outputs, which constitute a very complex response surface fitting problem. For this reason, we build a BPN to find out the response surface:

(1) Inputs: the five adjustable parameters – α, β, γ, η, and ϑ. All inputs and outputs are normalized into [0.1 0.9].
(2) Single hidden layer with 10 nodes.
(3) Outputs: the four objectives – the average cycle time, cycle time standard deviation, the maximum lateness, and the number of tardy jobs.
(4) Training algorithm: the Levenberg-Marquardt algorithm.
(5) Maximum number of epochs: 1000.

3. **Simulation Study.** To evaluate the effectiveness of the optimized-rule-fusion and certain-rule-first approach for multi-objective job scheduling in a wafer fabrication factory, we used simulated data to avoid disturbing the regular operations of the wafer fabrication factory [23,24].

To assess the effectiveness of the proposed methodology and to make comparison with some existing approaches – first in first out (FIFO), EDD, shortest remaining processing time (SRPT), CR, FSVCT, FSMCT, and nonlinear fluctuation smoothing rule (NFS) [24], all of these methods were applied to schedule the simulated wafer fabrication factory to collect the data of 1000 jobs, and then we separated the collected data by their product types and priorities. That is about the amount of work that can be achieved with 100% of the monthly capacity. In some cases, there was too little data, so they were not discussed. These rules are traditionally used for different purposes:

(1) The average cycle time: FIFO, FSMCT, SRPT, NFS.
(2) Cycle time standard deviation: FSVCT, NFS.
(3) The maximum lateness: EDD, CR.
(4) The number of tardy jobs: EDD, CR.

To determine the due date of a job, the FCM-BPN approach was applied to estimate the cycle time, for which the Levenberg-Marquardt algorithm rather than the gradient descent algorithm was applied to speed up the network convergence. Then, we added a constant allowance of three days to the estimated cycle time, i.e., $\kappa = 72$, to determine the internal due date.

Jobs with the highest priorities are usually processed first. In FIFO, jobs were sequenced on each machine first by their priorities, then by their arrival times at the machine. In EDD, jobs were sequenced first by their priorities, then by their due dates. In CR, jobs were sequenced first by their priorities, then by their critical ratios. In the proposed methodology, nine possible sets of the four adjustable parameters were first tried (see Table 2).

A fair way of comparison is to compare multiple performance measures at the same time [25-33]. Subsequently, the average cycle time, cycle time standard deviation, the number of tardy jobs, and the maximum lateness of all cases were calculated to assess the scheduling performance. With respect to the average cycle time, the FSMCT policy

TABLE 2. Some possible sets of the five adjustable factors

Model	$(\alpha, \beta, \gamma, \eta, \vartheta)$
Linear	(0.4, 0.2, 0.2, -0.2, 0.2), (0.6, 0.2, 0, -0.6, 0), (0.8, 0.2, -0.2, -1, -0.2), etc.
Nonlinear (u = 3; v = 3)	(0.016, 0.343, 0.064, 0.384, 0.064), (0.115, 0.001, 0.512, 0.685, 0.512), (0.681, 0.064, -0.008, -0.881, -0.008), etc.
Logarithmic	(0.3, 0.6, -0.43, -0.88, -0.43) , (0.5, 0.5, -0.41, -1, -0.41), (0.6, 0.3, -0.12, -0.84, -0.12), etc.

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	1254	400	317	1278	426
EDD	1094	345	305	1433	438
SRPT	948	350	308	1737	457
CR	1148	355	300	1497	440
FSMCT	1313	347	293	1851	470
FSVCT	1014	382	315	1672	475
NFS	1456	407	321	1452	421
The new rule $(0.4, 0.2, 0.2, -0.2, 0.2)$	1197	333	270	1351	368
The new rule $(0.6, 0.2, 0, -0.6, 0)$	1183	347	271	1160	339
The new rule $(0.8, 0.2, -0.2, -1, -0.2)$	1237	347	271	1089	344
The new rule $(0.016, 0.343, 0.064, 0.384, 0.064)$	922	307	267	1541	392
The new rule $(0.115, 0.001, 0.512, 0.685, 0.512)$	1070	309	260	1521	414
The new rule $(0.681, 0.064, -0.008, -0.881, -0.008)$	1230	348	270	1150	343
The new rule $(0.3, 0.6, -0.43, -0.88, -0.43)$	1143	353	274	1089	361
The new rule $(0.5, 0.5, -0.41, -1, -0.41)$	1206	355	280	1090	376
The new rule $(0.6, 0.3, -0.12, -0.84, -0.12)$	1191	344	278	1099	357

was used as the basis for comparison, while FSVCT was compared in evaluating cycle time standard deviation. On the other hand, the EDD and CR policies were used as the basis for comparison with respect to the maximum lateness and the number of tardy jobs, respectively. The results are summarized in Tables 3 $\sim$ 6.

According to the experimental results, the following points can be made:

(1) For the average cycle time, the proposed methodology outperformed the baseline approach, the FSMCT policy. The most obvious advantage was about 17% when $(\alpha, \beta, \gamma, \eta, \vartheta) = (0.6, 0.2, 0, -0.6, 0)$ or $(0.016, 0.343, 0.064, 0.384, 0.064)$.

(2) The proposed methodology also achieved a very good performance in reducing the maximum lateness, especially when the parameters were set to $(0.6, 0.2, 0, -0.6, 0)$ or $(0.6, 0.3, -0.12, -0.84, -0.12)$. Compared with EDD, the advantage was up to 31%.

Table 4. The performances of various approaches in the maximum lateness

The maximum lateness (hrs)	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
FIFO	401	−122	164	221	172
EDD	295	−181	144	336	185
SRPT	584	−142	174	718	194
CR	302	−159	138	423	192
FSMCT	875	−165	125	856	171
FSVCT	706	−112	174	686	260
NFS	627	10	161	331	151
The new rule $(0.4, 0.2, 0.2, -0.2, 0.2)$	524	−151	106	369	120
The new rule $(0.6, 0.2, 0, -0.6, 0)$	360	−152	118	21	94
The new rule $(0.8, 0.2, -0.2, -1, -0.2)$	753	−152	127	10	112
The new rule $(0.016, 0.343, 0.064, 0.384, 0.064)$	360	−170	112	449	155
The new rule $(0.115, 0.001, 0.512, 0.685, 0.512)$	521	−179	90	528	149
The new rule $(0.681, 0.064, -0.008, -0.881, -0.008)$	481	−164	95	34	92
The new rule $(0.3, 0.6, -0.43, -0.88, -0.43)$	661	−148	103	536	124
The new rule $(0.5, 0.5, -0.41, -1, -0.41)$	643	−150	103	137	100
The new rule $(0.6, 0.3, -0.12, -0.84, -0.12)$	396	−133	102	49	97

(3) In addition, the proposed methodology surpassed the FSVCT policy in reducing cycle time standard deviation. The most obvious advantage was 50% when the combination $(0.681, 0.064, -0.008, -0.881, -0.008)$ was used.

(4) In reducing the number of tardy jobs, the proposed methodology achieved the best performance when $(\alpha, \beta, \gamma, \eta, \vartheta) = (0.016, 0.343, 0.064, 0.384, 0.064)$ or $(0.3, 0.6, -0.43, -0.88, -0.43)$.

(5) As expected, SRPT performed well in reducing the average cycle times, especially for product types with short cycle times (e.g., product A), but might give an exceedingly bad performance with respect to cycle time standard deviation. If the cycle time is long, the remaining cycle time will be much longer than the remaining processing time, which leads to the ineffectiveness of SRPT. SRPT is similar to FSMCT. Both try to make all jobs equally early or late.

(6) The performance of EDD was also satisfactory for product types with short cycle time. If the cycle time is long, it is more likely to deviate from the prescribed internal due date, which leads to the ineffectiveness of EDD. That becomes more serious if the percentage of the product type is high in the product mix (e.g., product type A). CR has similar problems.

According to these results, the treatments carried out in this study did indeed improve the performances of the traditional policies. Subsequently, to optimize the scheduling performance of the proposed methodology, we built a BPN to find out the relationship between the scheduling performances (considering the largest case – product type A with normal priority) and the adjustable parameters. The Levenberg-Marquardt algorithm was

TABLE 5. The performances of various approaches in cycle time standard deviation

Cycle time standard deviation (hrs)	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
FIFO	55	24	25	87	51
EDD	129	25	22	50	63
SRPT	248	31	22	106	53
CR	69	29	18	58	53
FSMCT	419	33	16	129	104
FSVCT	280	37	27	201	77
NFS	87	49	19	44	47
The new rule $(0.4, 0.2, 0.2, -0.2, 0.2)$	294	43	23	126	35
The new rule $(0.6, 0.2, 0, -0.6, 0)$	71	41	22	30	29
The new rule $(0.8, 0.2, -0.2, -1, -0.2)$	127	33	23	56	31
The new rule $(0.016, 0.343, 0.064, 0.384, 0.064)$	259	31	16	78	72
The new rule $(0.115, 0.001, 0.512, 0.685, 0.512)$	320	38	12	142	66
The new rule $(0.681, 0.064, -0.008, -0.881, -0.008)$	89	35	15	50	32
The new rule $(0.3, 0.6, -0.43, -0.88, -0.43)$	215	35	18	245	46
The new rule $(0.5, 0.5, -0.41, -1, -0.41)$	140	35	16	96	54
The new rule $(0.6, 0.3, -0.12, -0.84, -0.12)$	90	35	13	77	38

applied to train the BPN. Only 10 epochs was required. The minimum MSE is $8.68 * 10^{-7}$. The results were summarized in Table 7. To construct the response surface, 100 points were randomly chosen.

The response surface is a 9-dimensional space. To visualize the response surface, it can be projected down to the 3-dimensional space. See Figure 1 for example.

From the 100 points on the response surface, the minima occurred at $(\alpha, \beta, \gamma, \eta, \vartheta) = (0.15, 0.60, 0.34, 0.59, -0.30)$ and was estimated to achieve the following scheduling performances:

The average cycle time (product type A, normal priority) = 931 (hrs)
The maximum lateness (product type A, normal priority) = 311 (hrs)
Cycle time standard deviation (product type A, normal priority) = 80 (hrs)
The number of tardy jobs (product type A, normal priority) = 55 (jobs)

Subsequently, a confirmation simulation study was carried out with this setting. The results are as follows:

The average cycle time (product type A, normal priority) = 953 (hrs)
The maximum lateness (product type A, normal priority) = 218 (hrs)
Cycle time standard deviation (product type A, normal priority) = 126 (hrs)
The number of tardy jobs (product type A, normal priority) = 71 (jobs)

The deviations from the estimated performances from the actual values are:

The average cycle time = 2%
The maximum lateness = 30%
Cycle time standard deviation = 37%

Table 6. The performances of various approaches in the number of tardy jobs

Cycle time standard deviation (hrs)	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
FIFO	79	0	12	16	5
EDD	71	0	12	19	5
SRPT	37	0	12	19	5
CR	79	0	12	19	5
FSMCT	58	0	12	19	5
FSVCT	56	0	12	18	5
NFS	79	1	12	19	5
The new rule $(0.4, 0.2, 0.2, -0.2, 0.2)$	67	0	12	19	5
The new rule $(0.6, 0.2, 0, -0.6, 0)$	79	0	12	19	5
The new rule $(0.8, 0.2, -0.2, -1, -0.2)$	79	0	12	19	5
The new rule $(0.016, 0.343, 0.064, 0.384, 0.064)$	52	0	12	19	5
The new rule $(0.115, 0.001, 0.512, 0.685, 0.512)$	59	0	12	19	5
The new rule $(0.681, 0.064, -0.008, -0.881, -0.008)$	79	0	12	19	5
The new rule $(0.3, 0.6, -0.43, -0.88, -0.43)$	79	0	12	12	5
The new rule $(0.5, 0.5, -0.41, -1, -0.41)$	79	0	12	17	5
The new rule $(0.6, 0.3, -0.12, -0.84, -0.12)$	79	0	12	18	5

Table 7. The estimated scheduling performances (product type A, normal priority)

#	α	β	γ	η	ϑ	The average cycle time	The maximum lateness	Cycle time standard deviation	The number of tardy jobs
1	0.029	0.225	0.460	−0.350	−0.288	951	312	135	51
2	0.033	0.382	0.095	0.219	−0.082	941	320	201	52
3	0.038	0.006	0.257	0.231	0.495	1121	521	329	55
4	0.049	0.493	−0.246	0.677	−0.219	1004	333	201	54
5	0.053	0.412	−0.237	0.424	0.340	1056	404	278	54
6	0.061	0.171	−0.155	−0.661	−0.247	1094	314	146	52
					...				
100	0.786	0.230	0.435	−0.893	0.159	1253	400	112	81

The number of tardy jobs = 23%

The advantages of the optimized rule over the traditional rules in the four aspects were 27%, 26%, 55%, and 10%, respectively.

4. **Conclusions and Directions for Future Research.** Multi-objective scheduling in a wafer fabrication factory is a challenging but important task. For such a complex production system, to optimize a single objective has been tough enough, needless to say taking into account four objectives at the same time. As an innovative attempt, this study presents an optimized-rule-fusion and certain-rule-first approach for multi-objective job

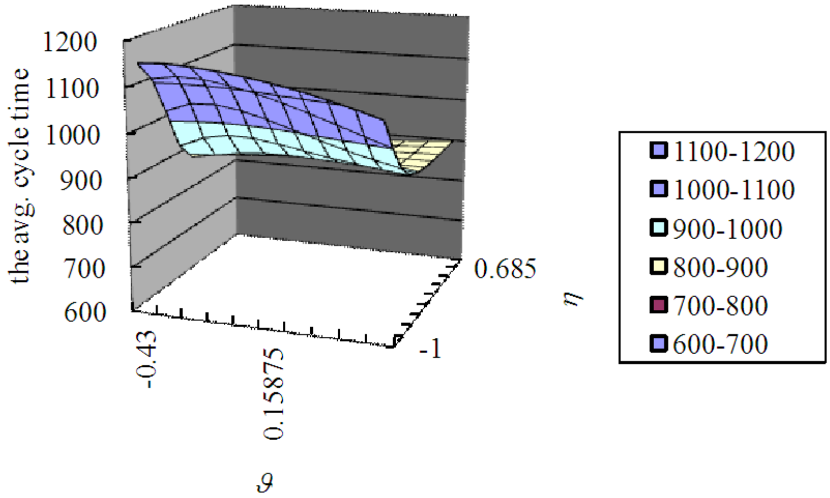

FIGURE 1. Projecting the response surface down to the 3-dimensional space ($\alpha = 0.433$; $\beta = 0.529$; $\gamma = -0.141$)

scheduling in a wafer fabrication factory, to optimize four performance measures at the same time, which has rarely been discussed in the past studies.

The optimized-rule-fusion and certain-rule-first approach proposes a nonlinear fusion of four traditional dispatching rules – EDD, FSMCT, FSVCT, and CR. However, in theory there is a contradiction among these rules, making their fusion a challenging task. To resolve this problem, a certain-rule-first procedure is established to retain the ranks of jobs with the highest or lowest priorities in the original rules. Some theoretical properties of the new rule are also proven. We also apply a more effective approach to estimate the remaining cycle time of a job, which is empirically shown to be conducive to the scheduling performance. Finally, the values of the adjustable parameters in the new rule are optimized using RSM instead of being chosen in a subjective way as in the previous studies.

After a simulation study, we observed the following phenomena:

(1) Through improving the accuracy of estimating the remaining cycle time, the performance of a scheduling rule can indeed be strengthened.
(2) In particular, the nonlinear way of rule fusion appears as an appropriate tool to analyze multi-objective scheduling problems. Such an advantage is further strengthened by using RSM to optimize the adjustable factors in the nonlinear fusion.
(3) Most variants of the new rule dominated most of the traditional rules compared in the experiment. Therefore, the proposed methodology can be concluded as an effective means to optimize the average cycle time, cycle time standard deviation, the maximum lateness, and the number of tardy jobs at the same time.

However, to further assess the effectiveness and efficiency of the proposed methodology, the only way is to apply it an actual wafer fabrication factory. In addition, different objectives can be fused in the same way in future studies.

Acknowledgment. This work is partially supported by the National Science Council of Taiwan.

REFERENCES

[1] T. Loukil, J. Teghem and D. Tuyttens, Solving multi-objective production scheduling problems using metaheuristics, *European Journal of Operational Research*, vol.161, no.1, pp.42-61, 2005.

[2] C. Grimme and J. Lepping, Combining basic heuristics for solving multi-objective scheduling problems, *2011 IEEE Symposium on Computational Intelligence in Scheduling*, pp.9-16, 2011.

[3] T. Chen, A tailored nonlinear fluctuation smoothing rule for semiconductor manufacturing factory scheduling, *Proc. of the Institution of Mechanical Engineers, Part I, Journal of Systems and Control Engineering*, vol.223, pp.149-160, 2009.

[4] L. N. van Wassenhove and F. Gelders, Solving a bicriterion scheduling problem, *European Journal of Operational Research*, vol.2, no.4, pp.281-290, 1980.

[5] C. Stein and J. Wein, On the existence of schedules that are nearoptimal for both makespan and total weighted completion time, *Operations Research Letters*, vol.21, no.3, pp.115-122, 1997.

[6] J. K. Cochran, S.-M. Horng and J. W. Fowler, A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines, *Computers & Operations Research*, vol.30, no.7, pp.1087-1102, 2003.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, 2001.

[8] C. A. C. Coello, G. B. Lamont and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer-Verlag, New Jersey, USA, 2007.

[9] T. Chen, Dynamic fuzzy-neural fluctuation smoothing rule for jobs scheduling in a wafer fabrication factory, *Proc. of the Institution of Mechanical Engineers, Part I, Journal of Systems and Control Engineering*, vol.223, pp.1081-1094, 2009.

[10] T. Chen, Fuzzy-neural-network-based fluctuation smoothing rule for reducing the cycle times of jobs with various priorities in a wafer fabrication factory – A simulation study, *Proc. of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture*, vol.223, pp.1033-1044, 2009.

[11] 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.

[12] 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.4289-4304, 2009.

[13] 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.

[14] T. Chen and Y. C. Wang, Incorporating the FCM-BPN approach with nonlinear programming for internal due date assignment in a wafer fabrication plant, *Robotics and Computer Integrated Manufacturing*, vol.26, pp.83-91, 2010.

[15] K. Pal and S. K. Pal, Soft computing methods used for the modelling and optimisation of Gas Metal Arc Welding: A review, *International Journal of Manufacturing Research*, vol.6, no.1, pp.15-29, 2011.

[16] T. Chen, Y.-C. Wang and H.-R. Tsai, Lot cycle time prediction in a ramping-up semiconductor manufacturing factory with a SOM-FBPN-ensemble approach with multiple buckets and partial normalization, *International Journal of Advanced Manufacturing Technology*, vol.42, no.11-12, pp.1206-1216, 2009.

[17] X. L. Xie and G. Beni, A validity measure for fuzzy clustering, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol.13, pp.841-847, 1991.

[18] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, Berlin, 2006.

[19] T. Chen and Y. C. Wang, A nonlinear scheduling rule incorporating fuzzy-neural remaining cycle time estimator for scheduling a semiconductor manufacturing factory, *International Journal of Advanced Manufacturing Technology*, vol.45, pp.110-121, 2009.

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

[21] M. L. Pinedo, *Scheduling Theory, Algorithm, and System*, Springer, New York, 2008.

[22] T.-C. Chiang and L.-C. Fu, Using a family of critical ratio-based approaches to minimize the number of tardy jobs in the job shop with sequence dependent setup times, *European Journal of Operational Research*, vol.196, pp.78-92, 2009.

[23] 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, pp.709-721, 2009.
[24] P. C. Chang, J. C. Hsieh and T. W. Liao, Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory, *Journal of Intelligent Manufacturing*, vol.16, pp.549-557, 2005.
[25] A. Thomas and P. Charpentier, Reducing simulation models for scheduling manufacturing facilities, *European Journal of Operational Research*, vol.161, pp.111-125, 2005.
[26] M. H. Zarandi and Z. S. Razaee, A fuzzy cluster-ing model for fuzzy data with outliers, *International Journal of Fuzzy System Applications*, vol.1, no.2, pp.29-42, 2011.
[27] P. Singamsetty and S. Panchumarthy, Automatic fuzzy parameter selection in dynamic fuzzy voter for safety critical systems, *International Journal of Fuzzy System Applications*, vol.2, no.2, pp.68-90, 2012.
[28] A. Al-Refaie and M. Li, Optimizing the performance of plastic injection molding using weighted additive model in goal programming, *International Journal of Fuzzy System Applications*, vol.1, no.2, pp.43-54, 2011.
[29] T. Chen, Applying a fuzzy and neural approach for forecasting the foreign exchange rate, *International Journal of Fuzzy System Applications*, vol.1, no.1, pp.36-48, 2011.
[30] S. Yang, K. Park and Z. Bien, Gesture spotting using fuzzy garbage model and user adaptation, *International Journal of Fuzzy System Applications*, vol.1, no.3, pp.47-65, 2011.
[31] R. R. Yager, On possibilistic and probabilistic information fusion, *International Journal of Fuzzy System Applications*, vol.1, no.3, pp.1-14, 2011.
[32] 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.
[33] T. Chen, A PCA-FBPN approach for job cycle time estimation in a wafer fabrication factory, *International Journal of Fuzzy System Applications*, vol.2, no.2, pp.50-67, 2012.