J Intell Manuf (2012) 23:897–911
DOI 10.1007/s10845-010-0445-9

Intelligent scheduling approaches for a wafer fabrication factory

Toly Chen

Received: 9 August 2009 / Accepted: 2 August 2010 / Published online: 15 August 2010

Abstract The production system of a wafer fabrication factory is a very complicated process. Job scheduling in a wafer fabrication factory is a very difficult task. To solve this problem, two intelligent scheduling rules are proposed in this study. The intelligent scheduling rules are modified from the well-known fluctuation smoothing rules with some innovative treatments. To evaluate the effectiveness of the proposed methodology, production simulation was also applied in this study. According to experimental results, the proposed methodology outperformed some existing approaches by reducing the average cycle time and cycle time standard deviation, the most important objectives of job scheduling in a wafer fabrication factory.

Keywords Intelligent · Wafer fabrication · Scheduling · Fuzzy · Neural

Introduction

The process of semiconductor manufacturing typically consists of four phases: wafer fabrication, wafer probe, assembly, and final testing. The production system of a wafer fabrication factory is a very complicated process with typical characteristics such as: fluctuating demand, jobs with various product types and priorities, un-balanced capacity, reentry of jobs into machines, hundreds of processing steps, alternative machines with unequal capacity, shifting bottlenecks, and others. This makes scheduling in a wafer fabrication factory a very difficult task. Pan and Chen (2004) studied the problem of scheduling reentrant job shops and flow shops with the objectives of minimizing makespan and mean flow time. Some heuristics were proposed in their study. Chen (2006) established a branch and bound procedure for the reentrant permutation flow-shop scheduling problem. Topaloglu and Kilincli (2009) proposed a modified shifting bottleneck heuristic (MSBH) for the reentrant job shop scheduling problem to minimize makespan. However, it is impossible to construct an optimization model and then to derive heuristics for a large-scale re-entrant production system like a wafer fabrication factory.

Wein (1998) demonstrated that a good job release policy leads to a significant improvement in the average cycle time, while many wafer fabrication factories (especially foundry factories) had to release the jobs associated with an order as soon as possible after the order is received. In addition, many studies have shown that applying general scheduling rules (such as first-in first out (FIFO), earliest due date (EDD), least slack (LS), shortest processing time (SPT), shortest remaining processing time (SRPT), critical ratio (CR), FIFO+, SRPT+, and SRPT++) to a wafer fabrication factory does not lead to very good results. Nevertheless, the research focusing on scheduling a wafer fabrication factory has become a very important issue at present (Gupta and Sivakumar 2006). In short, the existing approaches have the following problems:

(1) Most scheduling rules in this field consider only the attributes of the jobs gathered at the same place and lack an effective way of taking into consideration the conditions of the entire factory. However, in a wafer fabrication environment, we have access to real-time shop-floor status information for the entire factory.
(2) Most scheduling approaches are deterministic and do not reflect the changes in a wafer fabrication factory. Although there are a few scheduling rules that incorporates stochastic variables, such as the fluctuation

T. Chen (✉)
Department of Industrial Engineering and Systems Management,
Feng Chia University, Taichung City 407, Taiwan
e-mail: tolychen@ms37.hinet.net

smoothing (FS) rules, fluctuation smoothing policy for variance of cycle time (FSVCT) and fluctuation smoothing policy for mean cycle time (FSMCT), they use the average values for these stochastic variables, and in fact are not responsive to environmental changes. As a result, such a treatment leads to mis-scheduling.

(3) Most data-mining based approaches were developed for relatively small manufacturing systems like a job shop. In addition, these approaches attempt to simulate the best practices of the past for future scheduling applications. However, a wafer fabrication factory is a highly dynamic environment in which future conditions might be very different from those in the past. It is also very difficult to determine the so-called best practices for such a highly dynamic and complicated manufacturing system.
(4) Most scheduling rules are not tailored to a specific wafer fabrication factory.
(5) There is scant literature in which the historical data of a real wafer fabrication factory have been collected, since most studies in this field used simulated data.

To solve these problems and to further improve the performance of scheduling jobs in a wafer fabrication factory, two intelligent scheduling rules are proposed in this study. The intelligent scheduling rules are modifications from the well-known FS rules with the following treatments:

(1) Estimate the remaining cycle time with the self-organization map and fuzzy back propagation network (SOM-FBPN) approach (Chen and Wu 2008). Although "scheduling" is a hot topic, there are few published studies incorporating "remaining cycle time estimation" and "heuristic rule". Chang et al. (2009) proposed the hybrid SOM and case-based reasoning approach that can also be applied for this purpose.
(2) Normalize the components of the FS rules. Then apply the division operator instead of the traditional subtraction operator: The purpose is to enhance both the balancing (i.e. all parameters are of equal importance) and responsiveness (i.e. the correlation between any parameter and the slack is high) of the FS rules.
(3) Tailor the contents of the rules for the specific wafer fabrication factory with two adjustment factors.

With these innovative treatments, the intelligent scheduling rules are expected to achieve a better scheduling performance, as measured in terms of the average cycle time and cycle time standard deviation, in a wafer fabrication factory. It is obviously of great economic importance to reduce the average cycle time and cycle time standard deviation.

The remainder of this paper is organized as follows. A literature review is given in Sect. "Literature review". Section "Methodology" is divided into three parts. The first part describes the application of the SOM-FBPN approach to estimate the remaining cycle time of a job. Then the components of the FS rules are normalized and the division operator is applied instead of the traditional subtraction operator, in order to magnify the difference in the slack and to improve the responsiveness of the rules. Finally the rules are tailored to the wafer fabrication factory to be scheduled with two adjustment factors in the third part. To evaluate the effectiveness of the proposed methodology, production simulation is conducted in Sect. "Production simulation for generating test data". Based on the results of the analysis, some points are discussed in Sect. "Results and discussions". Finally, we draw our conclusions and provide some directions for future research in Sect. "Conclusion and directions for future research".

Literature review

Parameters that will be used in this study are defined as follows:

(1) R_n: the release time of job n.
(2) BQ_n: the total queue length before bottlenecks at R_n.
(3) CT_n: the cycle time (actual value) of job n.
(4) CTE_n: the estimated cycle time of job n.
(5) $D_n^{(i)}$: the delay of the i-th recently completed job, $i = 1 \sim 3$.
(6) FQ_n: the total queue length in the whole factory at R_n.
(7) Q_n: the total queue length on the processing route of job n at R_n.
(8) RCT_{nj}: the remaining cycle time (actual value) of job n at step j.
(9) $RCTE_{nj}$: the estimated remaining cycle time of job n at step j.
(10) SCT_{nj}: the step cycle time (actual value) of job n at step j.
(11) $SCTE_{nj}$: the estimated step cycle time of job n at step j.
(12) U_n: the average factory utilization at R_n.
(13) WIP_n: the factory WIP at R_n.

In addition, the fuzzy variable $\tilde{X}$ is derived by multiplying the importance of its crisp value X.

Kim et al. (2001) classified the problems of scheduling a wafer fabrication factory into three categories: job release control, job scheduling in serial processing workstations, and batch scheduling in batch processing workstations. There are two types of scheduling approaches: global approaches and local approaches. The intelligent scheduling approaches belong to the second category. Local scheduling approaches are usually focused on photolithography workstations, while global scheduling approaches can be applied to all workstations in a wafer fabrication factory. Gupta and Sivakumar (2006) classified the existing scheduling approaches for a wafer fabrication factory into four categories: heuristic rules, mathematical programming techniques, neighborhood search methods, and artificial intelligence techniques. Zhang et al. (2007) mentioned there are four classes of real-time scheduling approaches: semi-Markov decision modeling, dynamic programming, rule-based methods, and knowledge-based approaches. Most existing scheduling rules are "deterministic". Namely, the data used in these scheduling rules (e.g. the release time, the total processing time, and the due date of a job) do not change over time. Conversely, Lu et al. (1994) proposed two "stochastic" scheduling rules, FSVCT and FSMCT, in which the remaining cycle time of a job is considered and therefore needs to be estimated. The remaining cycle time of a job is highly stochastic because it is dependent not only on the factory conditions but also on the progresses of the other jobs (Chen 2003b). Other useful information when implementing the scheduling rules is the remaining processing time. Scheduling rules considering the remaining processing time include CR and LS. In these rules, the remaining processing time is known, while the remaining cycle time needs to be estimated. Theoretically, rules that consider the remaining cycle time are more effective than those considering the remaining processing time. Both scheduling rules have been shown to be effective in reducing the average cycle time and cycle time standard deviation. However, the problem is that the remaining cycle time is difficult to estimate. Another way of designing a stochastic scheduling rule is to combine some deterministic scheduling rules, and every time pick only the most suitable one. For example, Hsieh et al. (2001) used a combination of five scheduling rules including FSMCT, FSVCT, largest deviation first (LDF), one step ahead (OSA), and FIFO jointly. The problem with this approach is that, each time an extensive simulation experiment is required to estimate the performance of each candidate in order to determine the most suitable one. Nevertheless, Hsieh et al. addressed at a broad scope (including wafer release, short term scheduling, and rule composition) and proposed an intelligent use of efficient simulation for that scope.

Recently, some agent technologies have been applied in this field. Yoon and Shen (2008) constructed a multiple-agent system for scheduling a wafer fabrication factory, in which four types of agents (scheduling agents, workcell agents, machine agents, and product agents) were used. The optimal scheduling plan was found by the scheduling agent through enumerating a few possible scenarios. Their proposed methodology was only compared with the two basic scheduling rules FIFO and EDD. In addition, the common batch processing of jobs was not considered, and therefore the case was not proven to be practical. Recently, data-mining has been applied also in scheduling manufacturing systems. For example, in Li and Sigurdur (2004), historical schedules were transformed into appropriate data files that were mined in order to find out which past scheduling decisions corresponded to the best practices. Youssef et al. (2002) proposed a hybrid genetic algorithm (GA) and data mining approach to determine the optimal scheduling plan for a job shop, in which GA was used to generate a learning population of good solutions. These good solutions were then mined to find some decision rules that could be transformed into a meta-heuristic. Koonce and Tsai (2000) proposed a similar methodology. In addition, various priorities are assigned to jobs in a wafer fabrication factory, which has not explored in the past studies.

Methodology

Step 1: Estimating the remaining cycle time with the SOM-FBPN approach

The intelligent scheduling rules consist of three steps. First, the SOM-FBPN approach is applied to estimate the remaining cycle time of each job to be scheduled. With more accurate remaining cycle time estimation, the proposed methodology is expected to achieve a better scheduling performance.

In the SOM-FBPN approach, jobs to be scheduled are pre-classified into different categories with SOM. Other classification methods (e.g. k-means, fuzzy c-means) are also applicable. The structure of SOM is 10*10, and the number of output nodes is 100. Let x_n denotes the eight-dimensional feature vector $(U_n, Q_n, BQ_n, FQ_n, WIP_n, D_n^{(1)}, D_n^{(2)}, D_n^{(3)})$ corresponding to job n. The feature vectors of all jobs are fed into an SOM network with the following learning algorithm:

(1) Set the number of output nodes and the number of input nodes. Initialize the learning rate, the neighborhood size, and the number of iterations.
(2) Initialize the weights (w_{ij}) randomly where $i = 1 \sim p$ and p stands for the number of output nodes; $j = 1 \sim 8$.
(3) (Iteration) Provide an input vector to the network.
(4) Find the output node (winner) based on the similarity between the input vector and the weight vector. For an input vector x_n, the winning unit can be determined by distance $||x_n - w_c|| = \min_i ||x_n - w_i||$, where w_i is the weight vector of the i-th unit and the index c refers to the winning unit.
(5) Update the weight vector of the winner node using Kohonen's learning rule (Kohonen 1995).

$$w_i(t+1) = w_i(t) + \alpha(t)(x_n - w_i) \quad \text{for each } i \in N_c(t), \tag{1}$$

where t is the discrete-time index of the variables; the factor $\alpha(t) \in [0, 1]$ is a scalar that defines the relative size of the learning step; $N_c(t)$ specifies the neighborhood around the winner in the map array.

(6) Stop if the number of iterations has been completed. Otherwise, go to step 3.

Subsequently, examples of different categories are then learned with different FBPNs but with the same topology. Fuzzy approach have been applied to various research fields (e.g. Falda et al. 2010; Chen 2010). Shiue and Su (2002) constructed a neural network-based adaptive scheduling system for scheduling a flexible manufacturing system. Although a flexible manufacturing system considers the uncertainty in the manufacturing process, most flexible manufacturing systems are considerably smaller than a wafer fabrication factory. A back propagation network was constructed in Shiue and Su's study to capture the mapping between system state attributes and the dispatching rule under various performance criteria. The approach for scheduling the flexible manufacturing system was then chosen from some common dispatching rules.

The configuration of the FBPN is established as follows:

(1) Inputs: eight parameters associated with the n-th example/job including $U_n, Q_n, BQ_n, FQ_n, WIP_n$, and $D_n^{(i)} (i = 1 \sim 3)$. These parameters were most influential to the cycle time of a job according to the results of the backward elimination of regression analysis (Chen 2003a; Chang et al. 2005). These parameters are normalized and weighted as per Chen and Wu (2008).

(2) Single hidden layer: Generally one or two hidden layers are beneficial for the convergence property of the network.

(3) The number of neurons in the hidden layer is the same as in the input layer. This treatment has been adopted by many studies (e.g. Chen 2003a).

(4) Output: the (normalized) estimated cycle time (CTE_n) or the (normalized) step cycle time ($SCTE_{nj}$) of the example. In other words, there will be two groups of FBPNs. The first group is for estimating the CTE_n's of all the jobs to be scheduled, while the other group is for estimating their $SCTE_{nj}$'s. Then, the estimated remaining cycle time ($RCTE_{nj}$) can be derived in the following way:

$$RCTE_{nj} = \left(CTE_n - SCT_{nj}\right) * \left(1 + \log\left(SCT_{nj}/SCTE_{nj}\right)\right) \quad (2)$$

(5) Network learning rule: Delta rule.

(6) Transformation function: Sigmoid function,

$$f(x) = 1/(1 + e^{-x}). \quad (3)$$

(7) Learning rate (η) : $0.01 \sim 1.0$.

(8) Batch learning.

The procedure for determining parameter values is briefly described. After pre-classification, a portion of the adopted jobs in each category are fed into the FBPN as "training examples" to determine the parameter values for the category. The training stage consists of two phases. In the forward phase, inputs are multiplied with weights, added together, and transferred to the hidden layer. Then, activated signals from the hidden layer are transferred to the output layer using the same procedure. Finally, the output of the FBPN $\tilde{o}_n$ is generated. $\tilde{o}_n$ is defuzzified according to the centroid-of-area (COA) formula, and then, the defuzzification result o_n is compared with the actual value (the normalized cycle time or the normalized step cycle time) a_n to evaluate the accuracy of the FBPN which is represented with root-mean-squared-error (RMSE) or mean absolute percentage error (MAPE). 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 the step cycle time of the new job. In this study, a VB.NET program has been constructed to implement the FBPN.

Step 2: Improving the responsiveness of the FS rules

In traditional FS rules there are two different formulation methods, depending on the purpose. One way is aimed at minimizing the mean cycle time with the FSMCT rule (Lu et al. 1994):

$$SK_{nj} = n/\lambda - RCTE_{nj}. \quad (4)$$

where λ indicates the mean release rate. The other method is to minimize the variance of the cycle time with the FSVCT rule (Lu et al. 1994):

$$SK_{nj} = R_n - RCTE_{nj}. \quad (5)$$

In the traditional FSMCT rule, $RCTE_{nj}$ might be much greater than n/λ. As a result, the slack of a job becomes determined solely by $RCTE_{nj}$. To deal with this problem, both terms in the FSMCT rule are normalized as follows:

Table 1 A practical example ($\lambda = 0.026$)

n	R_n	n/λ	$RCTE_{nj}$
1	102	39	1399
2	163	77	647
3	197	116	743
4	208	154	530
5	457	193	810
6	469	232	1116
7	478	270	942
8	497	309	883
9	523	347	1851
10	596	386	2047
11	625	425	1036
12	652	463	1822
13	699	502	995
14	756	540	1127
15	783	579	2040
16	798	618	1146
17	800	656	2366
18	804	695	2092
19	826	733	1223
20	836	772	2151

$$Nor(RCTE_{nj}) = (RCTE_{nj} - \min_{\text{all } n}(RCTE_{nj}))/(\max_{\text{all } n}(RCTE_{nj}) - \min_{\text{all } n}(RCTE_{nj})), \quad (6)$$

$$\begin{aligned} Nor(n/\lambda) &= (n/\lambda - \min_{\text{all } n}(n/\lambda))/(\max_{\text{all } n}(n/\lambda) - \min_{\text{all } n}(n/\lambda)) \\ &= (n/\lambda - 1/\lambda)/(N/\lambda - 1/\lambda)) \\ &= (n-1)/(N-1). \end{aligned} \quad (7)$$

After normalization, both terms now range from 0 to 1.

Since the FS policies are based on differentiating SK_{nj}'s values, magnifying the differences in SK_{nj} seems to be a good way of enhancing the performance of the FS policy. To improve the balancing and responsiveness of the FSMCT rule, the division operator is applied instead of the traditional subtraction operator:

$$\text{Nonlinear FSMCT rule}: SK_{nj} = Nor(n/\lambda)/Nor(RCTE_{nj}) \quad (8)$$

To evaluate the balancing of the nonlinear FSMCT rule, the absolute value of the correlation coefficient $\rho_{x,y}$ is defined as follows:

$$\rho_{x,y} = \left(\sum xy - n\bar{x}\bar{y}\right) \Big/ \left(\sqrt{\sum x^2 - n\bar{x}^2} \cdot \sqrt{\sum y^2 - n\bar{y}^2}\right) \quad (9)$$

$\rho_{x,y}$ between SK_{nj} and $RCTE_{nj}$ can be calculated and compared with that between SK_{nj} and n/λ. Taking the practical data in Table 1 as an example. With the traditional FSMCT rule, the absolute value of the correlation coefficient between SK_{nj} and $RCTE_{nj}$ is 0.92, while that between SK_{nj} and n/λ is only 0.28. The deviation is up to 0.64 or 69% (over the larger value 0.92). Conversely, with the new rule, the absolute value of the correlation coefficient between SK_{nj} and $Nor(RCTE_{nj})$ is 0.34, while that between SK_{nj} and $Nor(n/\lambda)$ is 0.26. The deviation shrinks to only 0.08 or 23% over the larger value. Obviously, the balancing of the nonlinear FSMCT rule is much better than that of the traditional FSMCT rule.

On the other hand, the responsiveness of the nonlinear FSMCT rule can be evaluated with the absolute value of the coefficient of variation (CV) of SK_{nj},

$$|\text{CV}\left(SK_{nj}\right)| = |\delta_{SK_{nj}}/\mu_{SK_{nj}}, \quad (10)$$

which is equal to 4.46, while that of the traditional FSMCT rule is only 0.49. $\delta_{SK_{nj}}$ and $\mu_{SK_{nj}}$ denote the standard deviation and the average value of SK_{nj}, correspondingly. Obviously, the responsiveness can be improved with the nonlinear FSMCT rule. Similar analyses can be done on the nonlinear FSVCT rule:

$$\text{Nonlinear FSVCT rule}: SK_{nj} = Nor(R_n)/Nor(RCTE_{nj}), \quad (11)$$

where

$$Nor(R_n) = (R_n - \min_{\text{all } n}(R_n))/(\max_{\text{all } n}(R_n) - \min_{\text{all } n}(R_n)). \quad (12)$$

Step 3: Tailoring the nonlinear FS rules with two adjustment factors

Most scheduling rules in this field cannot be tailored to the wafer fabrication factory that is to be scheduled. To address this problem, the transition from a traditional FSMCT rule to its nonlinear form is analyzed as follows. The nonlinear form can be re-written as
Nonlinear FSMCT rule:

$$\begin{aligned} SK_{nj} &= Nor(n/\lambda)/Nor(RCTE_{nj}) \\ &= \frac{(n/\lambda - 1/\lambda)/(N/\lambda - 1/\lambda)}{(RCTE_{nj} - \min(RCTE_{nj}))/(\max(RCTE_{nj}) - \min(RCTE_{nj}))} \\ &= \frac{\beta}{\alpha} \cdot \frac{n/\lambda - 1/\lambda}{RCTE_{nj} - \min(RCTE_{nj})} \\ &= \frac{\beta}{\alpha} \cdot \frac{n/\lambda - 1/\lambda - RCTE_{nj} + RCTE_{nj}}{RCTE_{nj} - \min(RCTE_{nj})} \end{aligned}$$

Table 2 The performances of various approaches in reducing the average cycle time

Average cycle time (h)	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)	C (normal)	C (hot)
p-FIFO	1256	401	320	1278	457	1418	574
p-EDD-5.0	1087	346	306	1433	478	1755	585
p-EDD-5.5	1074	346	302	1464	464	1822	611
p-EDD-6.0	1047	350	298	1488	481	1863	590
p-EDD-6.5	1033	347	304	1556	484	1928	580
p-EDD-7.0	1022	353	302	1570	493	1945	592
p-EDD-7.5	1012	352	302	1593	476	1951	580
p-SRPT	966	350	309	1737	483	1971	580
p-FSMCT	1401	405	320	1408	430	1352	484
p-FSVCT	1046	385	317	1745	519	1884	606
TNFSMCT ($\xi = 0.25, \zeta = 0.5$)	1369	379	306	1361	399	1337	471
TNFSMCT ($\xi = 0.5, \zeta = 0.5$)	1337	384	295	1336	415	1329	510
TNFSMCT ($\xi = 0.585, \zeta = 0.5$)	1342	379	296	1336	397	1339	478
TNFSMCT ($\xi = 1, \zeta = 1$)	1353	379	298	1271	409	1253	496
TNFSMCT ($\xi = 1.322, \zeta = 1.5$)	1076	368	302	1660	462	1689	528
TNFSMCT ($\xi = 1.5, \zeta = 1.5$)	1371	404	299	1229	424	1271	491
TNFSMCT ($\xi = 2.25, \zeta = 1.5$)	1519	396	292	1338	400	1442	487
TNFSVCT ($\xi = 0.25, \zeta = 0.5$)	1437	371	291	1465	408	1375	477
TNFSVCT ($\xi = 0.5, \zeta = 0.5$)	1419	395	298	1413	418	1361	479
TNFSVCT ($\xi = 0.585, \zeta = 0.5$)	1417	393	308	1405	428	1419	477
TNFSVCT ($\xi = 1, \zeta = 1$)	1345	385	302	1460	434	1391	479
TNFSVCT ($\xi = 1.322, \zeta = 1.5$)	1087	354	307	1725	480	1639	553
TNFSVCT ($\xi = 1.5, \zeta = 1.5$)	1113	363	313	1721	479	1599	559
TNFSVCT ($\xi = 2.25, \zeta = 1.5$)	1567	379	304	1338	429	1430	479

$$= \frac{\beta}{\alpha(RCTE_{nj} - \min(RCTE_{nj}))} \cdot (n/\lambda - RCTE_{nj} + RCTE_{nj} - 1/\lambda) = \left(\frac{\alpha\left(RCTE_{nj} - \min(RCTE_{nj})\right)}{\beta}\right)^{-1} \cdot \left(n/\lambda - RCTE_{nj} + \left(RCTE_{nj} - 1/\lambda\right)\right) \cdot 1) \tag{13}$$

where $\alpha = N/\lambda - 1/\lambda$ and $\beta = \max(RCTE_{nj}) - \min(RCTE_{nj})$. Conversely, the linear form can also be rewritten as
Linear FSMCT rule:

$$SK_{nj} = n/\lambda - RCTE_{nj} = \left(\frac{\alpha(RCTE_{nj} - \min(RCTE_{nj}))}{\beta}\right)^{-0} \cdot (n/\lambda - RCTE_{nj} + (RCTE_{nj} - 1/\lambda) \cdot 0) \tag{14}$$

These two formulas can be generalized into the following form:

Tailored nonlinear FSMCT rule with two adjustment factors ξ and ζ, TNFSMCT(ξ, ζ):

$$SK_{nj} = 0.1 + 0.8 \cdot \left(\frac{\alpha\left(RCTE_{nj} - \min(RCTE_{nj})\right)}{\beta}\right)^{-\xi} \cdot \left(n/\lambda - RCTE_{nj} + (RCTE_{nj} - 1/\lambda) \cdot \zeta\right) \tag{15}$$

where ξ and ζ are positive real numbers satisfying the following constraints:

$$\text{If } \xi = 0 \text{ then } \zeta = 0, \text{ and vice versa} \tag{16}$$

$$\text{If } \xi = 1 \text{ then } \zeta = 1, \text{ and vice versa.} \tag{17}$$

There are many possible models to form the combinations of ξ and ζ. For example,

$$\text{Linear model}: \xi = \zeta \tag{18}$$

$$\text{Nonlinear model}: \xi = \zeta^k, k^3 0 \tag{19}$$

$$\text{Logarithmic model}: \xi = \ln(1 + \zeta)/\ln 2 \tag{20}$$

With any model, the proposed methodology tries various combinations of ξ and ζ to optimize the scheduling performance in the target wafer fabrication factory. In this way, the nonlinear FSMCT rule becomes tailored to the specific

Table 3 The performance of various approaches in reducing the cycle time standard deviation

Cycle time standard deviation (h)	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)	C (normal)	C (hot)
p-FSVCT	319	35	28	222	55	290	54
p-FIFO	56	24	23	87	40	72	31
p-EDD-5.0	130	25	23	50	39	134	23
p-EDD-5.5	103	34	17	60	28	147	60
p-EDD-6.0	101	31	22	41	49	144	34
p-EDD-6.5	90	25	20	38	53	141	36
p-EDD-7.0	83	24	13	35	47	141	34
p-EDD-7.5	74	30	17	42	42	151	33
p-SRPT	246	32	23	106	30	250	37
p-FSMCT	42	44	23	35	28	80	34
TNFSMCT ($\xi = 0.25, \zeta = 0.5$)	75	37	17	47	19	132	24
TNFSMCT ($\xi = 0.5, \zeta = 0.5$)	71	40	23	44	34	150	54
TNFSMCT ($\xi = 0.585, \zeta = 0.5$)	70	42	20	48	30	163	41
TNFSMCT ($\xi = 1, \zeta = 1$)	82	40	22	52	13	129	47
TNFSMCT ($\xi = 1.322, \zeta = 1.5$)	209	33	24	198	49	307	42
TNFSMCT ($\xi = 1.5, \zeta = 1.5$)	98	48	14	53	24	174	45
TNFSMCT ($\xi = 2.25, \zeta = 1.5$)	143	49	19	108	21	194	41
TNFSVCT ($\xi = 0.25, \zeta = 0.5$)	87	43	20	45	35	155	31
TNFSVCT ($\xi = 0.5, \zeta = 0.5$)	91	47	25	70	28	167	54
TNFSVCT ($\xi = 0.585, \zeta = 0.5$)	90	47	20	72	40	183	39
TNFSVCT ($\xi = 1, \zeta = 1$)	35	34	26	29	21	131	29
TNFSVCT ($\xi = 1.322, \zeta = 1.5$)	372	36	18	213	69	218	38
TNFSVCT ($\xi = 1.5, \zeta = 1.5$)	360	29	29	200	59	228	32
TNFSVCT ($\xi = 2.25, \zeta = 1.5$)	177	43	22	113	31	218	45

wafer fabrication factory. In addition, the values of ξ and ζ can be dynamically adjusted to reflect the changes in the production conditions of the wafer fabrication factory. On the other hand, the tailored nonlinear FSVCT rule with two adjustment factors can be obtained in a similar way:

Tailored nonlinear FSVCT rule with two adjustment factors ξ and ζ, TNFSVCT(ξ, ζ):

$$SK_{nj} = 0.1 + 0.8 \cdot \left(\frac{\alpha \left(RCTE_{nj} - \min(RCTE_{nj}) \right)}{\beta} \right)^{-\xi} \cdot \left(R_n - RCTE_{nj} + (RCTE_{nj} - \min(R_n)) \cdot \zeta \right) \quad (21)$$

To consider two performance measures simultaneously, TNFSMCT and TNFSVCT can be aggregated as follows:

Bi-criteria nonlinear fluctuation smoothing rule:

$$SK_{nj} = 0.1 + 0.8 \cdot \left(\frac{\alpha \left(RCTE_{nj} - \min(RCTE_{nj}) \right)}{\beta} \right)^{-\xi} \cdot \left(R_n - RCTE_{nj} + (RCTE_{nj} - \min(R_n)) \cdot \zeta \right) \cdot \left(n/\lambda - RCTE_{nj} + (RCTE_{nj} - 1/\lambda) \cdot \eta \right) \quad (22)$$

where ξ, ζ, and η are positive real numbers satisfying the following constraints:

$$\text{If } \zeta = 0 \text{ then } h = 0, \text{ and vice versa} \quad (23)$$

$$\xi = \zeta + h \text{ or } \max(\zeta, h) \quad (24)$$

Production simulation for generating test data

To generate test data, a simulation program using Microsoft Visual Basic .NET is developed to simulate a wafer fabrication environment with the following assumptions:

(1) Jobs are uniformly released into the factory. The simulated factory is a dynamic random access memory (DRAM) manufacturing factory. The uniform release policy is commonly adopted in such make-to-stock factories instead of the Poisson release policy. Improving the balancing of FS rules is considered especially important to a factory with the uniform release policy.
(2) The distribution of the interarrival times of machine downs is exponential.

Table 4 The improvement by each approach in reducing the average cycle time

Average cycle time (h)	A (normal) (%)	A (hot) (%)	A (super hot) (%)	B (normal) (%)	B (hot) (%)	C (normal) (%)	C (hot) (%)
p-FIFO	–	–	–	–	–	–	–
p-EDD-5.0	13	14	4	−12	−5	−24	−2
p-EDD-5.5	14	14	6	−15	−2	−28	−6
p-EDD-6.0	17	13	7	−16	−5	−31	−3
p-EDD-6.5	18	13	5	−22	−6	−36	−1
p-EDD-7.0	19	12	6	−23	−8	−37	−3
p-EDD-7.5	19	12	6	−25	−4	−38	−1
p-SRPT	23	13	3	−36	−6	−39	−1
p-FSMCT	−12	−1	0	−10	6	5	16
p-FSVCT	17	4	1	−37	−14	−33	−6
TNFSMCT ($\xi = 0.25, \zeta = 0.5$)	−9	5	4	−6	13	6	18
TNFSMCT ($\xi = 0.5, \zeta = 0.5$)	−6	4	8	−5	9	6	11
TNFSMCT ($\xi = 0.585, \zeta = 0.5$)	−7	5	8	−5	13	6	17
TNFSMCT ($\xi = 1, \zeta = 1$)	−8	5	7	1	11	12	14
TNFSMCT ($\xi = 1.322, \zeta = 1.5$)	14	8	6	−30	−1	−19	8
TNFSMCT ($\xi = 1.5, \zeta = 1.5$)	−9	−1	7	4	7	10	14
TNFSMCT ($\xi = 2.25, \zeta = 1.5$)	−21	1	9	−5	12	−2	15
TNFSVCT ($\xi = 0.25, \zeta = 0.5$)	−14	7	9	−15	11	3	17
TNFSVCT ($\xi = 0.5, \zeta = 0.5$)	−13	1	7	−11	9	4	17
TNFSVCT ($\xi = 0.585, \zeta = 0.5$)	−13	2	4	−10	6	0	17
TNFSVCT ($\xi = 1, \zeta = 1$)	−7	4	6	−14	5	2	17
TNFSVCT ($\xi = 1.322, \zeta = 1.5$)	13	12	4	−35	−5	−16	4
TNFSVCT ($\xi = 1.5, \zeta = 1.5$)	11	9	2	−35	−5	−13	3
TNFSVCT ($\xi = 2.25, \zeta = 1.5$)	−25	5	5	−5	6	−1	17

(3) The distribution of the time required to repair a machine is uniform.
(4) The percentages of jobs with different product types in the factory are predetermined. Therefore, this study is focused only on fixed-product-mix cases.
(5) The percentages of jobs with different priorities released into the factory are controlled.
(6) A job has equal chances to be processed on each alternative machine/head available at each step.
(7) A job cannot proceed to the next step until the fabrication on all its pieces has been finished.
(8) No preemption is allowed.

The basic configuration of the simulated wafer fabrication factory is simplified from a real-world wafer fabrication factory which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. Assumptions (1)∼(3), and (6)∼(8) are commonly adopted in related researches (e.g. Chen 2003a; Chen and Wu 2008), while assumption (5) is made to simplify the situation. There are more than ten products in the wafer fabrication factory. Only five major products (labeled as A∼E) are considered in the simulation model. The percentages of these products in the factory's product mix are assumed to be 35, 24, 17, 15, and 9%, respectively. The simulated wafer fabrication factory has a monthly capacity of 20,000 pieces of wafers with 100% utilization. Jobs are released into the wafer fabrication factory one by one every 0.85 h. Namely, the mean release rate $\lambda = 1/0.85 = 1.18$ jobs per hour. Three types of priorities (normal, hot, and super hot) are randomly assigned to jobs. The percentages of jobs with these priorities released into the wafer fabrication factory are restricted to be approximately 60, 30, and 10%, respectively. Each product has 150∼200 steps and 6∼9 re-entrances to the most bottlenecked machine. A total of 102 machines (including alternative machines) are provided to process single-wafer or batch operations in the wafer fabrication factory. The simulation model is not only large and complicated but also capable of demonstrating the characteristics of the real wafer fabrication factory. The conclusions drawn here are therefore meaningful to the control of the real wafer fabrication factory.

Thirty simulation replications are run successively. The time required for each simulation replication is about 15 min on a PC with 256MB RAM and Athlon^{TM} 64 Processor 3000+ CPU. A horizon of 24 months is simulated.

Table 5 The improvement by each approach in reducing the cycle time standard deviation

Cycle time standard deviation (h)	A (normal) (%)	A(hot) (%)	A (super hot) (%)	B (normal) (%)	B (hot) (%)	C (normal) (%)	C (hot) (%)
p-FSVCT	–	–	–	–	–	–	–
p-FIFO	82	31	18	61	27	75	43
p-EDD-5.0	59	29	18	77	29	54	57
p-EDD-5.5	68	3	39	73	49	49	−11
p-EDD-6.0	68	11	21	82	11	50	37
p-EDD-6.5	72	29	29	83	4	51	33
p-EDD-7.0	74	31	54	84	15	51	37
p-EDD-7.5	77	14	39	81	24	48	39
p-SRPT	23	9	18	52	45	14	31
p-FSMCT	87	−26	18	84	49	72	37
TNFSMCT ($\xi = 0.25, \zeta = 0.5$)	76	−6	39	79	65	54	56
TNFSMCT ($\xi = 0.5, \zeta = 0.5$)	78	−14	18	80	38	48	0
TNFSMCT ($\xi = 0.585, \zeta = 0.5$)	78	−20	29	78	45	44	24
TNFSMCT ($\xi = 1, \zeta = 1$)	74	−14	21	77	76	56	13
TNFSMCT ($\xi = 1.322, \zeta = 1.5$)	34	6	14	11	11	−6	22
TNFSMCT ($\xi = 1.5, \zeta = 1.5$)	69	−37	50	76	56	40	17
TNFSMCT ($\xi = 2.25, \zeta = 1.5$)	55	−40	32	51	62	33	24
TNFSVCT ($\xi = 0.25, \zeta = 0.5$)	73	−23	29	80	36	47	43
TNFSVCT ($\xi = 0.5, \zeta = 0.5$)	71	−34	11	68	49	42	0
TNFSVCT ($\xi = 0.585, \zeta = 0.5$)	72	−34	29	68	27	37	28
TNFSVCT ($\xi = 1, \zeta = 1$)	89	3	7	87	62	55	46
TNFSVCT ($\xi = 1.322, \zeta = 1.5$)	−17	−3	36	4	−25	25	30
TNFSVCT ($\xi = 1.5, \zeta = 1.5$)	−13	17	−4	10	−7	21	41
TNFSVCT ($\xi = 2.25, \zeta = 1.5$)	45	−23	21	49	44	25	17

The maximal cycle time is less than 3 months. Therefore, 4 months and an initial WIP status (obtained from a pilot simulation run) are sufficient to drive the simulation into a steady state. The statistical data were collected starting at the end of the fourth month. For each replication, data of 30 jobs were collected and classified by their product types and priorities. In total, the data of 900 jobs were collected. A trace report was generated every simulation run for verifying the simulation model. Simulated average cycle times were compared with actual values to validate the simulation model.

Results and discussions

To evaluate the effectiveness of the proposed methodology and to make a comparison with some existing approaches—p-FIFO, p-EDD, p-SRPT, p-FSVCT, and p-FSMCT, all these methods were applied to schedule the simulated wafer fabrication factory so as to collect the data of 900 jobs that were then separated by their product types and priorities. In total, the data of 7*900 = 6300 jobs were collected. Subsequently, the average cycle time and cycle time standard deviation of jobs with every product type and priority were calculated to evaluate the scheduling performance. The manner in which the competition for machines by jobs is resolved in a wafer fabrication factory has a clear bearing on these two performance measures. The results are summarized in Tables 2 and 3. The percentages of improvement by each approach are shown in Tables 4 and 5.

In p-FIFO, jobs were sequenced on each machine first by their priorities, then by their arrival times at the machine. In p-EDD, jobs were sequenced first by their priorities, then by their due dates. The performance of p-EDD is dependent on the way of determining the due date of a job. In the experiment, the due date of a job was determined as follows:

$$\text{Due date} = \text{release time} + (\psi - 1.5 * \text{priority}) * \text{total processing time} \tag{25}$$

where ψ indicates the cycle time multiplier.

In p-FSVCT and p-FSMCT, there were two stages. First, jobs were scheduled with the p-FIFO policy, for which the

Table 6 Some feasible combinations of ξ and ζ

Rule	(ξ, ζ)
Linear	(0, 0), (0.5, 0.5), (1, 1), (1.5, 1.5), etc.
Nonlinear ($k = 2$)	(0, 0), (0.25, 0.5), (1, 1), (2.25, 1.5), etc.
Logarithmic	(0, 0), (0.585, 0.5), (1, 1), (1.322, 1.5), etc.

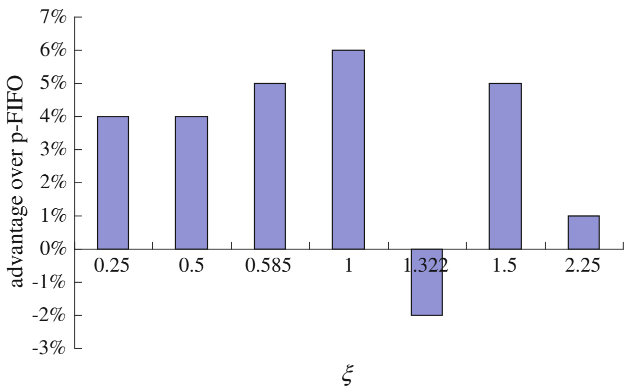

Fig. 1 The effect of ξ on reducing the average cycle times

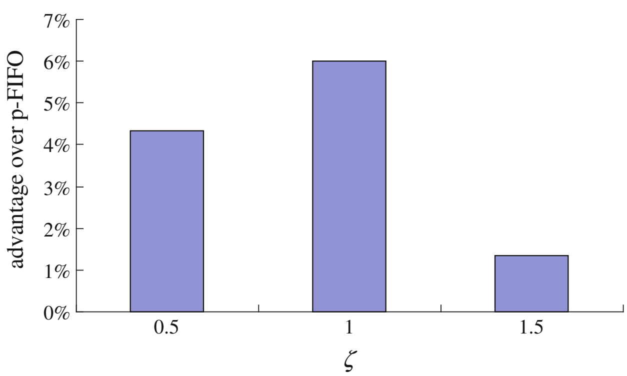

Fig. 2 The effect of ζ on reducing the average cycle times

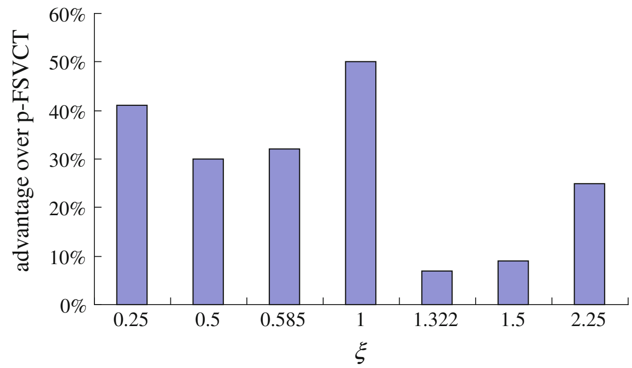

Fig. 3 The effect of ξ on reducing the cycle time standard deviations

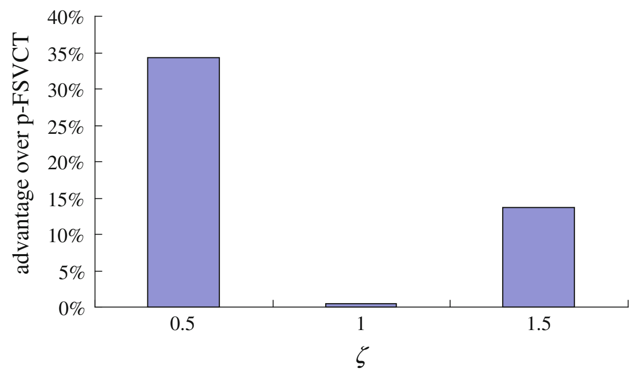

Fig. 4 The effect of ζ on reducing the cycle time standard deviations

remaining cycle times at each step of all jobs were recorded and averaged. Then, the p-FSVCT/p-FSMCT policy was applied to schedule jobs based on the average remaining cycle times obtained previously. In other words, jobs were sequenced on each machine first by their priorities, then by their slack values, which was equal to their release times minus the average remaining cycle times.

In the proposed methodology, the eight combinations of ξ and ζ in Table 6 were tried. Note that when they are equal to 0, the proposed methodology is equivalent to the traditional FS rules.

With respect to the average cycle time, the p-FIFO policy was adopted as the basis for comparison. For the cycle time standard deviation, the p-FSVCT policy was adopted as the basis for comparison.

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

(1) With respect to the average cycle time, the proposed TNFSMCT rule outperformed the baseline approach, the p-FIFO policy, in most cases. The most obvious advantage occurred when $(\xi, \zeta) = (1, 1)$, where the TNFSMCT rule surpassed the p-FIFO policy in 6 out of 7 cases with an average advantage of 6%. Putting more stress on n/λ in TNFSMCT than in FSMCT seems to be beneficial to the scheduling performance in a wafer fabrication factory with the uniform release policy.
(2) At the same time, the proposed TNFSVCT rule also achieved very good performances in reducing the cycle time standard deviations. The most obvious advantage happened when $(\xi, \zeta) = (1, 1)$, where the TNFSVCT rule surpassed the baseline p-FSVCT policy significantly in all cases with an average advantage of up to 50%. In TNFSVCT, the weights of the release time and the remaining cycle time are better balanced. Experimental results revealed that such a treatment did indeed reduce the fluctuation in the cycle time and improve the performance of the traditional FSVCT policy.
(3) The TNFSMCT rule also performed very well with respect to the cycle time standard deviation. The

most obvious advantage happened when $(\xi, \zeta) = (0.25, 0.5)$. The TNFSMCT rule reduced the cycle time standard deviation by 52% on average.

(4) The performances of the TNFSMCT rule using various values of ξ with respect to the average cycle time for several cases are compared in Fig. 1. We noticed that when ξ was equal to 1.322, the performance of the proposed methodology was not satisfactory. The reason is discussed as follows. The proposed methodology is equal to the traditional (linear) FSMCT rule when ξ is equal to 0. As ξ increases, the proposed methodology becomes more and more nonlinear. One problem of the traditional (linear) FSMCT is that the slack is determined solely by the remaining cycle time. For this reason, in the nonlinear FSMCT rule, the weight put on the remaining cycle time should be reduced. Therefore, according to Eq. (15) it is better to choose a small value of $-\xi/\zeta$. Among the tested combinations (except $\xi = 0.25$ that was not nonlinear enough), when ξ was equal to 1.322 the value of $-\xi/\zeta$ was the largest, which led to the poor performance of the proposed methodology in this case. On the other hand, the effects of ζ on reducing the average cycle time are shown in Fig. 2. The performances of the TNFSVCT rule using various values of ξ in reducing the cycle time standard deviations for several cases are compared in Fig. 3. On the other hand, the effect of ζ on reducing the cycle time standard deviations is shown in Fig. 4.

(5) The traditional p-FSVCT policy performed poorly in the simulation experiment. This might be due to the diversification in the product type and priority that made the remaining cycle time of a job highly uncertain and very difficult to estimate. As a result, estimating with the average value might be far from accurate and may impair the scheduling performance of the traditional p-FSVCT policy.

(6) As expected, p-SRPT performed well in reducing the average cycle times, but might give an exceedingly bad performance with respect to the cycle time standard deviation.

(7) Among the various EDD rules, the performance of p-EDD-5.0 was the best in reducing the average cycle times, while p-EDD-7.0 was the best choice if the cycle time standard deviations were to be minimized.

(8) The effects of the three treatments have been analyzed. Taking product type B (normal priority) as an example, the results are shown in Table 7. First, the SOM-FBPN approach was also applied to the traditional FSVCT rule. We noticed that with better remaining cycle time estimation, the performances of the traditional FSVCT rule were indeed improved. In addition, incorporating the SOM-FBPN approach with the nonlinear scheduling rule could reduce the cycle time standard deviation further. After tailoring the content of the new rule for the target wafer fabrication factory, such advantages became more obvious.

(9) To statistically compare the performance of these approaches in all cases, we first ranked them in each case, and then added up the ranks of the same approach for comparison. The results are summarized in Table 8. To consider both aspects, the performances of all approaches were compared in Table 9, which supported the Pareto optimality of the proposed methodology because most of its variants were not dominated by any of the traditional approaches. Namely, the performances of the proposed methodology in both respects were not simultaneously inferior to those of any other approach. In addition, TNFSMCT ($\xi = 0.25, \zeta = 0.5$) and TNFSVCT ($\xi = 1, \zeta = 1$) dominated all traditional approaches. On the other hand, p-FSVCT was dominated by the other approaches. Namely, the performances of p-FSVCT in both respects were worse than those of any other approach. The p-SRPT policy was dominated by the other approaches except p-FSVCT and TNFSVCT ($\xi = 0.5, \zeta = 0.5$), while the p-FIFO policy was only dominated by TNFSMCT ($\xi = 0.25, \zeta = 0.5$) and TNFSVCT ($\xi = 1, \zeta = 1$).

(10) In the experiment, TNFSMCT even surpassed TNFSVCT in reducing cycle time standard deviation. The possible reasons are discussed as follows. The traditional FSVCT rule attempts to make every job equally late or equally early, thereby reducing the standard deviation of lateness. Therefore, FSVCT or TNFSVCT is especially effective when all jobs have equal priorities. Conversely, in the simulation model, jobs had various priorities, and it became very difficult to make them equally late or equally early, which led to the poor performance of TNFSVCT in this case. Nevertheless, TNFSVCT still outperformed the five existing approaches in reducing cycle time variation.

(11) As stated above, only some values of ξ and ζ are feasible. The performance of the three models for generating (ξ, ζ) combinations are compared in Table 10. In this experiment, the nonlinear model seems to be the best choice for this purpose, which implies that a two-factor adjustment is better than a single-factor adjustment in which two parameters are equal.

To ascertain whether there were significant differences between the performances of the proposed scheduling rules and those of traditional scheduling rules, we applied Wilcoxon sign-rank test to test the following hypotheses:

H_{a0}: The performance of the proposed TNFSMCT ($\xi = 0.25, \zeta = 0.5$) is the same as those of traditional scheduling rules in the average cycle time respect.

Table 7 The effects of the three treatments

Approach	Average cycle time (h)	Cycle time standard deviation (h)
p-FSVCT	1745	319
p-FSVCT + SOM-FBPN	1448 (−17%)	163 (−27%)
Nonlinear + SOM-FBPN	1370 (−21%)	31 (−86%)
Tailored + nonlinear + SOM-FBPN	1229 (−30%)	29 (−91%)

Table 8 The sum of the ranks of each approach

Sum of ranks	Average cycle time	Total rank #	Cycle time standard deviation	Total rank #
p-FIFO	99	#18	56	#5
p-EDD-5.0	92	#15	67	#8
p-EDD-5.5	88	#11	83	#13
p-EDD-6.0	91	#14	75	#11
p-EDD-6.5	98	#17	65	#7
p-EDD-7.0	104	#20	46	#3
p-EDD-7.5	92	#15	57	#6
p-SRPT	110	#23	105	#18
p-FSMCT	101	#19	55	#4
p-FSVCT	135	#24	147	#24
TNFSMCT ($\xi = 0.25, \zeta = 0.5$)	60	#4	42	#1
TNFSMCT ($\xi = 0.5, \zeta = 0.5$)	57	#3	86	#14
TNFSMCT ($\xi = 0.585, \zeta = 0.5$)	44	#1	77	#12
TNFSMCT ($\xi = 1, \zeta = 1$)	52	#2	71	#9
TNFSMCT ($\xi = 1.322, \zeta = 1.5$)	90	#13	130	#23
TNFSMCT ($\xi = 1.5, \zeta = 1.5$)	70	#6	91	#15
TNFSMCT ($\xi = 2.25, \zeta = 1.5$)	77	#8	104	#17
TNFSVCT ($\xi = 0.25, \zeta = 0.5$)	63	#5	74	#10
TNFSVCT ($\xi = 0.5, \zeta = 0.5$)	76	#7	112	#19
TNFSVCT ($\xi = 0.585, \zeta = 0.5$)	89	#12	102	#16
TNFSVCT ($\xi = 1, \zeta = 1$)	80	#9	44	#2
TNFSVCT ($\xi = 1.322, \zeta = 1.5$)	105	#21	122	#21
TNFSVCT ($\xi = 1.5, \zeta = 1.5$)	109	#22	124	#22
TNFSVCT ($\xi = 2.25, \zeta = 1.5$)	83	#10	117	#20

H_{a1}: The performance of the proposed TNFSMCT ($\xi = 0.25, \zeta = 0.5$) is better than those of traditional scheduling rules in the average cycle time respect.

H_{b0}: The performance of the proposed TNFSVCT ($\xi = 1, \zeta = 1$) is the same as those of traditional scheduling rules in the cycle time standard deviation respect.

H_{b1}: The performance of the proposed TNFSVCT ($\xi = 1, \zeta = 1$) is better than those of traditional scheduling rules in the cycle time standard deviation respect.

H_{c0}: The performance of the proposed TNFSMCT ($\xi = 0.25, \zeta = 0.5$) is the same as those of traditional scheduling rules in the cycle time standard deviation respect.

H_{c1}: The performance of the proposed TNFSMCT ($\xi = 0.25, \zeta = 0.5$) is better than those of traditional scheduling rules in the cycle time standard deviation respect.

H_{d0}: The performance of the proposed TNFSVCT ($\xi = 1, \zeta = 1$) is the same as those of traditional scheduling rules in the average cycle time respect.

H_{d1}: The performance of the proposed TNFSVCT ($\xi = 1, \zeta = 1$) is better than those of traditional scheduling rules in the average cycle time respect.

The results of hypothesis testing are summarized in Table 11. The performance of the proposed TNFSMCT ($\xi = 0.25, \zeta = 0.5$) rule was significantly better than

Table 9 The results of Pareto analysis

#	Approach	Dominated by #
1	p-FIFO	11, 21
2	p-EDD-5.0	11, 21
3	p-EDD-5.5	11, 13, 14, 18, 21
4	p-EDD-6.0	11, 14, 18, 21
5	p-EDD-6.5	7, 11, 21
6	p-EDD-7.0	11, 21
7	p-EDD-7.5	11, 21
8	p-SRPT	1–7, 9, 11–14, 16–18, 20–21
9	p-FSMCT	11, 21
10	p-FSVCT	1–9, 11–24
11	TNFSMCT ($\xi = 0.25, \zeta = 0.5$)	None
12	TNFSMCT ($\xi = 0.5, \zeta = 0.5$)	13, 14
13	TNFSMCT ($\xi = 0.585, \zeta = 0.5$)	None
14	TNFSMCT ($\xi = 1, \zeta = 1$)	None
15	TNFSMCT ($\xi = 1.322, \zeta = 1.5$)	3, 11–14, 16–21, 24
16	TNFSMCT ($\xi = 1.5, \zeta = 1.5$)	11–14, 18
17	TNFSMCT ($\xi = 2.25, \zeta = 1.5$)	11–14, 16, 18
18	TNFSVCT ($\xi = 0.25, \zeta = 0.5$)	11, 14
19	TNFSVCT ($\xi = 0.5, \zeta = 0.5$)	11–14, 16, 18
20	TNFSVCT ($\xi = 0.585, \zeta = 0.5$)	3, 11–14, 16, 18, 21
21	TNFSVCT ($\xi = 1, \zeta = 1$)	11
22	TNFSVCT ($\xi = 1.322, \zeta = 1.5$)	1–7, 9, 11–14, 16–21, 24
23	TNFSVCT ($\xi = 1.5, \zeta = 1.5$)	1–7, 9, 11–14, 16–22, 24
24	TNFSVCT ($\xi = 2.25, \zeta = 1.5$)	11–14, 16–19, 21

the traditional FSMCT rule in the cycle time respect. On the other hand, the advantage of the proposed TNFSVCT ($\xi = 1, \zeta = 1$) rule over p-EDD-5.5, p-EDD-6.0, p-SRPT, and p-FSVCT with respect to the cycle time standard deviation was also statistically significant. Further, TNFSMCT ($\xi = 0.25, \zeta = 0.5$) also outperformed three traditional rules in reducing the cycle time standard deviations.

Table 10 The performances of the various models

	Sum of the ranks in all cases	
	Average cycle time	Cycle time standard deviation
Linear	615	531
Nonlinear	586	486
Logarithmic	631	519

Conclusion and directions for future research

Two intelligent scheduling approaches, TFSMCT and TFSVCT, were proposed in this study to further improve the performance of scheduling jobs in a wafer fabrication factory. The intelligent scheduling approaches were modified from the well-known FS rules with three innovative treatments. First, the remaining cycle time of a job was estimated by applying the SOM-FBPN approach to improve the estimation accuracy. Second, the components of the FS rules were normalized, and then the division operator was applied instead of the traditional subtraction operator to enhance the responsiveness of the rule. Third, the content of the intelligent scheduling rules can be tailored for the wafer fabrication factory to be scheduled with two adjustment factors. TFSMCT and TFSVCT generalize the FS rules by relaxing the constraints on the orders in the slack formulae.

To evaluate the effectiveness of the proposed methodology and to compare it with some existing approaches, production simulation was also conducted in this study, and then the proposed methodology and some existing approaches were all applied to schedule the simulated wafer fabrication factory. The experimental results were as follows:

Table 11 The results of testing hypotheses using Wilcoxon sign-rank test

	H_{a0}	H_{b0}	H_{c0}	H_{d0}
p-FIFO	$W = 12$	$W = 12$	$W = 14$	$W = 12$
p-EDD-5.0	8	8	6	11
p-EDD-5.5	9	3*	2*	10
p-EDD-6.0	9	3*	4	10
p-EDD-6.5	9	4	6	8
p-EDD-7.0	9	8	12	8
p-EDD-7.5	9	4	9	8
p-SRPT	8	3*	1**	8
p-FSMCT	0**	8	10	6
p-FSVCT	5	0**	1**	6

$* \ p < 0.05$
$** \ p < 0.025$
$*** \ p < 0.01$

(1) The proposed TNFSMCT rule outperformed the traditional approaches in reducing the average cycle time and the cycle time standard deviation at the same time.
(2) The proposed TNFSVCT rule outperformed the traditional approaches in reducing the cycle time standard deviation.
(3) To tailor the content of the intelligent scheduling rules, two-factor adjustment was shown to be better than single-factor adjustment.
(4) Considering both the average cycle time and the cycle time standard deviation, the proposed methodology was also a Pareto optimal solution.

However, to further evaluate the effectiveness and efficiency of the proposed methodology, it has to be applied to various types of wafer fabrication factories including foundry factories. Also, the content of the intelligent scheduling rules could be optimized through other methods such as the response surface methodology (RSM) (Zhang et al. 2007). Further, whether the proposed TNFSMCT and TNFSVCT are still effective under various release policies and loading levels needs to be examined in future studies.

References

Chang, P. C., Fan, C. Y., & Wang, Y. W. (2009). Evolving CBR and data segmentation by SOM for flow time prediction in semiconductor manufacturing. *Journal of Intelligent Manufacturing, 20*(4), 421–429.

Chang, P. C., Hsieh, J. C., & Liao, T. W. (2005). Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory. *Journal of Intelligent Manufacturing, 16*, 549–557.

Chen, J.-S. (2006). A branch and bound procedure for the reentrant permutation flow-shop scheduling problem. *International Journal of Advanced Manufacturing Technology, 29*(11–12), 1186–1193.

Chen, T. (2003a). A fuzzy back propagation network for output time prediction in a wafer fab. *Applied Soft Computing, 2/3F*, 211–222.

Chen, T. (2003b). A fuzzy mid-term single-fab production planning model. *Journal of Intelligent Manufacturing, 14*, 273–285.

Chen, T. (2008). A SOM-FBPN-ensemble approach with error feedback to adjust classification for wafer-lot completion time prediction. *International Journal of Advanced Manufacturing Technology, 37*(7–8), 782–792.

Chen, T. (2010). Applying a fuzzy and neural approach for forecasting the foreign exchange rate. *International Journal of Fuzzy System Applications*, (in press).

Chen, T., & Wu, H.-C. (2008). A fuzzy-neural approach for output projection in a semiconductor fabrication factory. *Journal of Chinese Institute of Engineers, 32*(2), 287–293.

Falda, M., Rossi, F., & Venable, K. B. (2010). Dynamic consistency of fuzzy conditional temporal problems. *Journal of Intelligent Manufacturing, 21*, 75–88.

Gupta, A. K., & Sivakumar, A. I. (2006). Job shop scheduling techniques in semiconductor manufacturing. *International Journal of Advanced Manufacturing Technology, 27*, 1163–1169.

Hsieh, B.-W., Chen, C.-H., & Chang, S.-C. (2001). Scheduling semiconductor wafer fabrication by using ordinal optimization-based simulation. *IEEE Transactions on Robotics and Automation, 17*(5), 599–608.

Kim, Y. D., Kim, J. G., & Kim, H. U. (2001). Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. *IEEE Transactions on Robotics and Automation, 17*(5), 589–598.

Kohonen, T. (1995). *Self-organizing maps*. Heidelberg: Springer.

Koonce, D. A., & Tsai, S.-C. (2000). Using data mining to find patterns in genetic algorithm solutions to a job shop schedule. *Computers and Industrial Engineering, 38*(3), 361–374.

Li, X., & Sigurdur, O. (2004). Data mining for best practices in scheduling data. In *Proceedings of IIE annual conference and exhibition*.

Lu, S. C. H., Ramaswamy, D., & Kumar, P. R. (1994). Efficient scheduling policies to reduce mean and variation of cycle time in semiconductor manufacturing plant. *IEEE Transactions on Semiconductor Manufacturing, 7*(3), 374–388.

Pan, J. C.-H., & Chen, J.-S. (2004). A comparative study of schedule-generation procedures for the reentrant shops scheduling problem. *International Journal of Industrial Engineering: Theory Applications and Practice, 11*(4), 313–321.

Shiue, Y. R., & Su, C. T. (2002). Attribute selection for neural network-based adaptive scheduling systems in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology, 20*(7), 532–544.

Topaloglu, S., & Kilinci, G. (2009). A modified shifting bottleneck heuristic for the reentrant job shop scheduling problem with makespan minimization. *International Journal of Advanced Manufacturing Technology, 44*(7–8), 781–794.

Wein, L. M. (1998). Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing, 1*, 115–130.

Yoon, H. J., & Shen, W. (2008). A multiagent-based decision-making system for semiconductor wafer fabrication with hard temporal constraints. *IEEE Transactions on Semiconductor Manufacturing, 21*(1), 83–91.

Youssef, H., Brigitte, C.-M., Noureddine, Z. (2002). 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* (vol. 7, pp. 280–285).

Zhang, H., Jiang, Z., & Guo, C. (2007). Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology. *International Journal of Advanced Manufacturing Technology, 41*(1–2), 110–121.