Int J Adv Manuf Technol (2007) 35:575–586
DOI 10.1007/s00170-006-0741-x

ORIGINAL ARTICLE

A hybrid look-ahead SOM-FBPN and FIR system for wafer-lot-output time prediction and achievability evaluation

Toly Chen

Received: 19 April 2006 / Accepted: 3 August 2006 / Published online: 22 November 2006

Abstract A hybrid system is constructed in this study for wafer-lot-output time prediction and achievability evaluation, which are critical tasks for a wafer fab (fabrication plant). In the first part of the hybrid system, a look-ahead self-organization map fuzzy-back-propagation network (SOM-FBPN) is constructed to predict the output time of a wafer lot. Compared with traditional approaches in this field, the look-ahead SOM-FBPN has three advanced features: incorporating the future release plan, classifying wafer lots, and incorporating expert opinions. According to experimental results, the prediction accuracy and efficiency of the look-ahead SOM-FBPN were significantly better than those of many existing approaches. In the second part of the hybrid system, a set of fuzzy inference rules (FIRs) are established to evaluate the achievability of an output time forecast, which is defined as the possibility that the fabrication on the wafer lot can be finished in time before the output time forecast. Achievability is as important as accuracy and efficiency but has been ignored in traditional studies. With the proposed methodology, both output time prediction and achievability evaluation can be concurrently accomplished.

Keywords Output time prediction · Wafer fab · Self-organization map · Fuzzy back propagation network · Fuzzy inference rules · Achievability

T. Chen (✉)
Department of Industrial Engineering and Systems Management,
Feng Chia University,
No. 100, Wenhwa Rd., Seatwen,
Taichung, Taiwan
e-mail: tolychen@ms37.hinet.net

1 Introduction

Predicting the output time for every lot in a wafer fab is a critical task not only to the fab itself, but also to its customers. After the output time of each lot in a wafer fab is accurately predicted, several managerial goals (including internal due-date assignment, output projection, ordering decision support, enhancing customer relationship, and guiding subsequent operations) can be simultaneously achieved [6]. Predicting the output time of a wafer lot is equivalent to estimating the cycle (flow) time of the lot, because the former can be easily derived by adding the release time (a constant) to the latter.

There are six major approaches commonly applied to predicting the output/cycle time of a wafer lot: multiple-factor linear combination (MFLC), production simulation (PS), back propagation networks (BPN), case-based reasoning (CBR), fuzzy modeling methods, and hybrid approaches. Among the six approaches, MFLC is the easiest, quickest, and most prevalent in practical applications. The major disadvantage of MFLC is the lack of forecasting accuracy [6]. Conversely, a huge amount of data and lengthy simulation time are two disadvantages of PS. Nevertheless, PS is the most accurate output-time prediction approach if the related databases are continuingly updated to maintain enough validity, and it often serves as a benchmark for evaluating the effectiveness of another method. Considering both effectiveness and efficiency, Chang et al. [5] and Chang and Hsieh [3] both forecasted the output/cycle time of a wafer lot with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy measured with the root mean squared error (RMSE) was considerably improved with these BPNs. For example, an improvement of about 40% in the RMSE was achieved in Chang et al. [5]. On the other

hand, much less time and fewer data are required to generate an output time forecast with a BPN than with PS. Chang et al. [4] proposed a *k*-nearest-neighbors based case-based reasoning (CBR) approach which outperformed the BPN approach in forecasting accuracy. In one case, the advantages were up to 27%. Chang et al. [5] modified the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the fuzzy modeling method proposed by Wang and Mendel [19], called the WM method, with a simple genetic algorithm (GA) and proposed the evolving fuzzy rule (EFR) approach to predict the cycle time of a wafer lot. Their EFR approach outperformed CBR and BPN in prediction accuracy. Chen [6] constructed a fuzzy BPN (FBPN) that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN especially in respect to efficiency.

According to these results, the concept of classifying inputs, which has been adopted in CBR and EFR, can indeed improve the effectiveness (prediction accuracy) of wafer-lot-output time prediction. This fact provides a motive for proposing a similar approach—a SOM classifier and then a FBPN regression for the same purpose. On the other hand, all the aforementioned methods are based on the historical data of the fab. However, a lot of studies have shown that the performance of sequencing and scheduling in a fab relies heavily on the future release plan, which has been neglected in this field. In addition, the characteristic reentrant production flows of a fab lead to the phenomenon that a lot that will be released in the future might appear in front of another lot that currently exists in the fab. For these reasons, to further improve the accuracy of wafer-lot-output time prediction, the future release plan of the fab has to be considered (look-ahead). As a result, a look-ahead SOM-FBPN is constructed to predict the output time of a wafer lot. Compared with traditional approaches in this field, the look-ahead SOM-FBPN has three advanced features:

1. The future release plan of the fab is incorporated
2. Wafer lots are classified
3. Expert opinions are incorporated

PS is also applied in this study to generate test examples. Using simulated data, the effectiveness and efficiency of the look-ahead SOM-FBPN are shown and compared with those of many existing approaches. On the other hand, traditional studies in this field are focused on accuracy and efficiency aspects. Another concept that is as important but has been ignored, is the "achievability" of an output time forecast, which is defined as the possibility that the fabrication on the wafer lot can be finished in time before the output time forecast. Theoretically, if a probability distribution can be obtained for the output time forecast, then the achievability can be evaluated with the cumulative probability of the probability distribution before the given date. However, there are many managerial actions (e.g. elevating the priority of the wafer lot, lowering the priority of another wafer lot, inserting emergency lots, adding allowance, etc.) that are more influential to the achievability. Considering their effects, the evaluation of the achievability is decomposed into the following two assessments: the possible forwardness of the output time forecast if the priority is elevated, and the ease of priority elevation. For combining the two assessments, the fuzzy AND operator is applied, followed by the establishment of a set of FIR facilitate the application. Finally, a hybrid look-ahead SOM-FBPN and FIR system is constructed to enhance the effectiveness and efficiency of wafer-lot-output time prediction, and to evaluate the achievability of an output time forecast.

2 Previous related work

As mentioned previously, predicting the output time of a wafer lot is equivalent to estimating the cycle time of the wafer lot. There are six major approaches commonly applied to predicting the output/cycle time of a wafer lot:

1. MFLC: The cycle time of a lot is estimated with the weighted sum of parameters includingthe following three points:
   - Job properties: The total processing time, the number of reentrances, and the number of operations of the lot
   - Cycle time and waiting time series: the actual cycle times, the waiting times, the total processing times, the numbers of reentrances, and the numbers of operations of some (usually three) of the most recently completed lots
   - Workload information: the number of jobs (work-in-progress, WIP) in the fab or waiting for the most bottlenecked machines or on the processing route of the lot, the average fab utilization

Many internal due-date setting rules belong to MFLC (see Table 1). Among the six approaches, MFLC is the easiest, quickest, and most prevalent in practical applications. The major disadvantage of MFLC is the lack of forecasting accuracy.

2. PS: A fab production simulation system continuously updating the related databases to maintain enough validity can also be applied to predicting/simulating the output time of a wafer lot (e.g. [16, 20]). Theoretically, a number of replicates of a probabilistic simulation need to be run to sufficiently consider all uncertain or stochastic properties and events (e.g.

Table 1 Some internal due-date setting rules

Method	Formula	Symbol meanings
Total work content (TWK)	$CT_n=\omega_1 NP_n+\omega_2 TP_n, OT_n=CT_n+RT_n.$	CT_n: cycle time of lot n
Number of operations (NOP) [1]		RT_n: release time of lot n
		NP_n: number of operations of lot n
		TP_n: total processing time of lot n
		OT_n: output time forecast of lot n
		ω_j: constants, for all j.
Current jobs-in-queue (JIQ) [18]	$CT_n = \omega_1 Q_n + \omega_2 NP_n + \omega_3 TP_n,$ $OT_n = CT_n + RT_n.$	Q_n: total queue length on the route of lot n
Cycle time sampling	$CT_n = \omega_1 \cdot \frac{\sum_{i=1}^{k} \frac{CT_{(i)}}{NP_{(i)}}}{k} \cdot NP_n + \omega_2 NP_n + \omega_3 TP_n, OT_n = CT_n + RT_n.$	$CT_{(i)}$: cycle time of the$i$-th most recently finished lot
Cycle time statistics referencing		$NP_{(i)}$: number of operations of the i-th most recently finished lot.
Operation flowtime sampling (OFS) [18]	$CT_n = \omega_1 \cdot \frac{\sum_{i=1}^{k} \frac{D_{(i)}}{NP_{(i)}}}{k} \cdot NP_n + \omega_2 TP_n, OT_n = CT_n + RT_n.$	
Average delay-in-queue (DIQ) [18]		TR$_{(i)}$: total processing time of the i-th most recently finished lot
		$D_{(i)}$: delay of the$i$-th most recently finished lot=$CT_{(i)}-TR_{(i)}$
Congestion and cycle time sampling	$CT_n = \omega_1 \cdot Q_n + \omega_2 \cdot \frac{\sum_{i=1}^{k} \frac{CT_{(i)}}{NP_{(i)}}}{k} \cdot NP_n + \omega_3 NP_n + \omega_4 TP_n, OT_n = CT_n + RT_n.$	
Congestion and operation flowtime sampling (COFS) [1]		
Exponential smoothing [11]	$CT_n = TP_n + D_n, D_n$ is continuously modified as follows: New $D_n = D_n + \alpha\left(D_{(1)} - D_n\right),$ $OT_n = CT_n + RT_n.$	α:: constant, $0\leq\alpha\leq1$
Empirical queuing approach [13]	$CT_n = TP_n + D_n(U),$ $D_n(U)$: the relationship (obtained by regression) between a lot's delay and the fab's utilization, $OT_n = CT_n + RT_n.$	
Bayesian cycle time prediction	CT_n(WIP, c) is fitted and modified after a Bayesian analysis, $c = \frac{r_b T_0}{r_b T_0 - 1}\left(\frac{\overline{CT}}{T_0} - 1\right), OT_n = CT_n(WIP, c) + RT_n.$	WIP: fab WIP
		c: congestion level
		r_b: bottleneck machine processing rate (jobs/unit time)
		T_0: the mean cycle time when WIP=1.
		$\overline{CT}$: the mean cycle time

inconsistent human-assisted operations, unexpected machine downs, etc.), so as to obtain a more reliable forecast. There are two shortages of PS: (1) huge amount of data needs to be maintained; (2) simulation time is often lengthy. Nevertheless, PS is the most accurate output time prediction approach (if the related databases are continuously updated to maintain enough validity), and often serves as a benchmark for evaluating the effectiveness of another method. PS also tends to be preferred because it allows for computational experiments and subsequent analyses without any actual execution [4].

3. BPN: Many studies have shown that artificial neural networks (ANN) outperform traditional methods in time-series forecasting [9]. The advantages of a BPN include the tolerance of noises [15], the speed of application, and the capability of simulating complex systems (such as a wafer fab). Chang and Liao [5] and Chang and Hsieh [3] both forecasted the output/cycle time of a wafer lot with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy (measured with the RMSE) was considerably improved with these BPNs. On the other hand, much less time and fewer data are required to

generate an output time forecast with a BPN than with PS.

4. CBR: Chang et al. [4] proposed a k-nearest-neighbors-based CBR approach with dynamic factor weights and a nonlinear similarity function for due-date assignment in a wafer fab, in which the weights of factors (the cycle times of the previous cases/lots) are proportional to the similarities of the new lot with the previous cases. Chang et al.'s CBR approach outperformed the BPN approach in forecasting accuracy.
5. Fuzzy modeling methods: Chang et al. [5] modified the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the WM method with a simple genetic algorithm and proposed the EFR approach to predict the cycle time of a wafer lot. Their EFR approach outperformed CBR and BPN in prediction accuracy. Genetic techniques have shown to be capable of carrying out a comprehensive optimization of the parameters.
6. Hybrid approaches: Chen [6] constructed a FBPN that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN in respect to efficiency. In regards to prediction accuracy measured with the minimal RMSE, the performance of the FBPN was slightly better than that of the BPN.

Traditional studies in this field are focused on accuracy and efficiency aspects. However, whether an output time forecast can be achieved or not has not been investigated, and that might be much more important from a managerial and practical viewpoint. As a summary of this section, a trade-off table for selecting output time prediction approaches is shown in Table 2.

3 Methodology

In this paper, a hybrid look-ahead SOM-FBPN and FIR system is constructed for lot-output time prediction and achievability evaluation in a wafer fab (see Fig. 1). The hybrid system is composed of two parts. In the first part, a look-ahead SOM-FBPN (see Fig. 2) is proposed to predict the output time of a wafer lot.

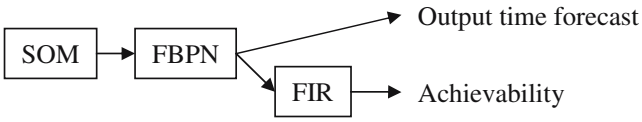

Fig. 1 The system architecture

3.1 Wafer-lot-output time prediction with a look-ahead SOM-FBPN

There are three steps in applying the look-ahead SOM-FBPN to predicting the output time of a wafer lot.

3.1.1 Step 1: incorporating the future release plan of the fab (look-ahead)

There are many possible ways to incorporate the future release plan in predicting the output time of a wafer lot currently existing in the fab. In this study, the three nearest future discounted workloads on the lot's processing route (according to the future release plan) are proposed for this purpose:

1. The first nearest future discounted workload ($FDW^{(1)}$): the sum of the (processing time/release time)'s of the operations of the lots that will be released within time (now, now+T1)
2. The second nearest future discounted workload ($FDW^{(2)}$): the sum of the (processing time/release time)'s of the operations of the lots that will be released within time (now+T1, now+ T1+T2)
3. The third nearest future discounted workload ($FDW^{(3)}$): the sum of the (processing time/release time's) of the operations of the lots that will be released within time (now+T1+T2, now+T1+T2+T3).

Note that only the operations performed on the machines on the lot's processing route are considered in calculating these future workloads, which then become three additional inputs to the FBPN.

3.1.2 Step 2: example classification with SOM: rationale and procedure

At first, the rationale for combining SOM and FBPN for wafer-lot output time prediction is explained as follows. Theoretically, a well-trained BPN or FBPN (without being

Table 2 A trade-off table for selecting output time prediction approaches

Approach	Data required	Execution time	Accuracy	Easy to use	Prevalence
PS	Huge	Lengthy	Very high	Easy	Not prevalent
MFLC	Small	Very short	Low	Very easy	prevalent
BPN, FBPN	Small	Short	High	Difficult	Not prevalent
CBR	Small	Short	High	Difficult	Not prevalent
EFR	Small	Short	High	Difficult	Not prevalent

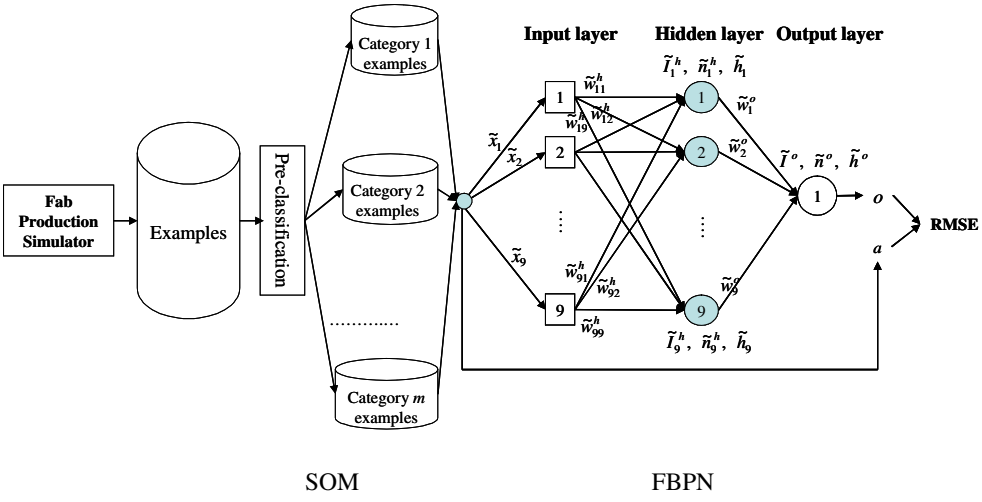

Fig. 2 The look-ahead SOM-FBPN

stuck to local minima) with a good selected topology can successfully map any complex distribution. However, wafer-lot output time prediction is a much more complicated problem, and the results of many previous studies (e.g. [4–6]) have shown the incapability of BPN or FBPN in solving such a problem. One reason is that there might be multiple complex distributions to model, and these distributions might be quite different (even for the same product type and priority). For example, when the workload level (in the wafer fab or on the processing route or before bottlenecks) is stable, the cycle time of a wafer lot basically follows the well-known Little's law [14], and the output time of the wafer lot can be easily predicted. Conversely, if the workload level fluctuates or keeps going up (or down), predicting the cycle time and output time of a wafer lot becomes much more difficult. For this reason, classifying wafer lots under different circumstances seems to be a reasonable treatment. In this respect, SOM can serve as a clustering tool for high dimensional data (e.g. production data in a wafer fab) [12]. Besides, there are several studies that suggested the hybrid use of SOM and BPN or FBPN (e.g. [2, 7, 17]), and the latter is a well-known approach to wafer-lot-output time prediction. As a result, the combination of SOM and FBPN is chosen in this study.

The procedure of applying SOM in forming inputs to the FBPN is now detailed. Every lot fed into the FBPN is called an example. Examples are pre-classified into different categories before they are fed into the FBPN with SOM. Let x_n denotes the nine-dimensional $(U_n, Q_n, BQ_n, FQ_n, WIP_n, D_n^{(i)}, FDW_n^{(1)}, FDW_n^{(2)}, \text{and } FDW_n^{(3)})$ feature vector corresponding to lot n. The feature vectors of all lots 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~m and m stands for the maximum number of classes (wafer lot categories), j=1~9.
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. $w_i(t+1) = w_i(t) + \alpha(t)(x_n - w_i)$ for each $i \in N_c(t)$, 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. Provide the next input vector and go to step 4. Otherwise, go to step 7.
7. Stop if the number of iterations has been completed. Otherwise, go to step 3.

After the training is accomplished, input vectors that are topologically close are mapped to the same category, and then the classification result is post-processed, including eliminating isolated examples, merging small blocks, etc. Finally, the classification is finished, and the value of 'm' is determined. After classification, examples of different categories are then learned with different FBPNs but with the same topology. The procedure for determining the parameter values is described in the next section.

3.1.3 Step 3: output time prediction within each category with FBPN

After pre-classification, a portion of the adopted examples in each category is fed as "training examples" into the

FBPN to determine the parameter values for the category. The configuration of the FBPN is established as follows:

1. Inputs: nine parameters associated with the n-th example/lot including the average fab utilization (U_n), the total queue length on the lot's processing route (Q_n) or before bottlenecks (BQ_n) or in the whole fab (FQ_n), the fab WIP (WIP$_n$), the latenesses $\left(D_n^{(i)}\right)$ of the$i$-th recently completed lots, and the three nearest future discounted workloads on the lot's processing route (FDW$^{(1)}$, FDW$^{(2)}$, and FDW$^{(3)}$). These parameters have to be normalized so that their values fall within (0, 1). Then some production execution/control experts are requested to express their beliefs (in linguistic terms) about the importance of each input parameter in predicting the cycle (output) time of a wafer lot. Linguistic assessments for an input parameter are converted into several pre-specified fuzzy numbers. The subjective importance of an input parameter is then obtained by averaging the corresponding fuzzy numbers of the linguistic replies for the input parameter by all experts. The subjective importance obtained for an input parameter is multiplied to the normalized value of the input parameter. After such a treatment, all inputs to the FBPN become triangular fuzzy numbers, and the fuzzy arithmetic for triangular fuzzy numbers is applied to deal with all calculations involved in training the FBPN.
2. Single hidden layer: Generally one or two hidden layers are more beneficial for the convergence property of the FBPN.
3. Number of neurons in the hidden layer: the same as that in the input layer. Such a treatment has been adopted by many studies (e.g. [4]). Two other formulas for determining the suitable number of neurons in the hidden layer are also listed for reference:
   - Number of neurons in the hidden layer=(number of neurons in the input layer + number of neurons in the output layer)/2.
   - Number of neurons in the hidden layer=(number of neurons in the input layer × number of neurons in the output layer)1/2.
4. Output: the (normalized) cycle time forecast of the example.
5. Network learning rule: delta rule.
6. Transformation function: Sigmoid function,
   $f(x) = \dfrac{1}{1+e^{-x}}$.
7. Learning rate (η): 0.01~1.0.
8. Batch learning.

The procedure for determining the parameter values is now described. After pre-classification, a portion of the adopted examples in each category is fed as "training examples" into the FBPN to determine the parameter values for the category. 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:

$$\widetilde{h}_j = \left(h_{j1}, h_{j2}, h_{j3}\right) = \frac{1}{1+e^{-\widetilde{n}_j^h}} = \left(\frac{1}{1+e^{-n_{j1}^h}}, \frac{1}{1+e^{-n_{j2}^h}}, \frac{1}{1+e^{-n_{j3}^h}}\right),$$

where

$$\widetilde{n}_j^h = \left(n_{j1}^h, n_{j2}^h, n_{j3}^h\right) = \widetilde{I}_j^h(-)\widetilde{\theta}_j^h = \left(I_{j1}^h - \theta_{j3}^h,\ I_{j2}^h - \theta_{j2}^h,\ I_{j3}^h - \theta_{j1}^h\right),$$
$$\widetilde{I}_j^h = \left(I_{j1}^h, I_{j2}^h, I_{j3}^h\right) = \sum_{all\ i} \widetilde{w}_{ij}^h(\times)\widetilde{x}_{(i)} \cong \left(\sum_{all\ i} \min\left(w_{ij1}^h x_{(i)1}, w_{ij3}^h x_{(i)3}\right), \sum_{all\ i} w_{ij2}^h x_{(i)2}, \sum_{all\ i} \max\left(w_{ij1}^h x_{(i)1}, w_{ij3}^h x_{(i)3}\right)\right),$$

and (–) and (×) denote fuzzy subtraction and multiplication, respectively; $\widetilde{h}_j$'s are also transferred to the output layer with the same procedure. Finally, the output of the FBPN is generated as:

$$\widetilde{o} = (o_1, o_2, o_3) = \frac{1}{1+e^{-\widetilde{n}^o}} = \left(\frac{1}{1+e^{-n_1^o}}, \frac{1}{1+e^{-n_2^o}}, \frac{1}{1+e^{-n_3^o}}\right),$$

where

$$\widetilde{n}^o = \left(n_1^o, n_2^o, n_3^o\right) = \widetilde{I}^o(-)\widetilde{\theta}^o = \left(I_1^o - \theta_3^o,\ I_2^o - \theta_2^o,\ I_3^o - \theta_1^o\right),$$
$$\widetilde{I}^o = \left(I_1^o, I_2^o, I_3^o\right) = \sum_{all\ j} \widetilde{w}_j^o(\times)\widetilde{h}_j \cong \left(\sum_{all\ j} \min\left(w_{j1}^o h_{j1},\ w_{j3}^o h_{j3}\right), \sum_{all\ j} w_{j2}^o h_{j2}, \sum_{all\ j} \max\left(w_{j1}^o h_{j1},\ w_{j3}^o h_{j3}\right)\right).$$

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

$$o = \mathrm{COA}(\widetilde{o}) = \frac{o_1 + 2o_2 + o_3}{4}.$$

Then the output o is compared with the normalized actual cycle time a, for which the RMSE is calculated:

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

Subsequently in the backward phase, the deviation between o and a is propagated backward, and the error

terms of neurons in the output and hidden layers can be calculated respectively as:

$$\begin{aligned}
&\delta^o = o(1-o)(a-o),\\
&\widetilde{\delta}_j^h = (\delta_{j1}^h, \delta_{j2}^h, \delta_{j3}^h) = \widetilde{h}_j(\times)(1-\widetilde{h}_j)(\times)\widetilde{w}_j^o\delta^o\\
&\quad \cong (\min(h_{j1})(1-h_{j3})w_{j1}^o,\ h_{j3}(1-h_{j1})w_{j1}^o)\delta^o,\\
&\qquad \max(h_{j3}(1-h_{j1})w_{j3}^o,\ h_{j1}(1-h_{j3})w_{j3}^o)\delta^o),\\
&\qquad h_{j2}(1-h_{j2})w_{j2}^o\delta^o,\ \max(\min(h_{j1}(1-h_{j3})w_{j1}^o,\\
&\qquad h_{j3}(1-h_{j1})w_{j1}^o)\delta^o,\ \max(h_{j3}(1-h_{j1})w_{j3}^o, h_{j1}(1-h_{j3})w_{j3}^o)\delta^o)).
\end{aligned}$$

Based on them, adjustments that should be made to the connection weights and thresholds can be obtained as:

$$\begin{aligned}
\Delta\widetilde{w}_j^o &= (\Delta w_{j1}^o, \Delta w_{j2}^o, \Delta w_{j3}^o) = \eta\delta^o\widetilde{h}_j\\
&= \eta\delta^o(\min(h_{j1},\ h_{j3}),\ h_{j2},\ \max(h_{j1},\ h_{j3})),\\
\Delta\widetilde{w}_{ij}^h &= (\Delta w_{ij1}^h,\ \Delta w_{ij2}^h,\ \Delta w_{ij3}^h) = \eta\widetilde{\delta}_j^h(\times)\widetilde{x}_i\\
&\cong \eta(\min(\delta_{j1}^h x_{i1},\ \delta_{j1}^h x_{i3},\ \delta_{j3}^h x_{i1},\ \delta_{j3}^h x_{i3}), \delta_{j2}^h x_{i2},\\
&\quad \max(\delta_{j1}^h x_{i1},\ \delta_{j1}^h x_{i3},\ \delta_{j3}^h x_{i1},\ \delta_{j2}^h x_{i3})),\\
\Delta\theta^o &= -\eta\delta^o,\\
\Delta\widetilde{\theta}_j^h &= (\Delta\theta_{j1}^h,\ \Delta\theta_{j2}^h,\ \Delta\theta_{j3}^h) = -\eta\widetilde{\delta}_j^h\\
&= (-\eta\delta_{j3}^h,\ -\eta\delta_{j2}^h,\ -\eta\delta_{j1}^h).
\end{aligned}$$

To accelerate convergence, a momentum can be added to the learning expressions. For example,

$$\begin{aligned}
\Delta\widetilde{w}_j^o &= \eta\delta^o\widetilde{h}_j + \alpha(\widetilde{w}_j^o(t) - \widetilde{w}_j^o(t-1))\\
&= (\eta\delta^o h_{j1} + \alpha w_{j1}^o(t) - \alpha w_{j3}^o(t-1),\\
&\quad \eta\delta^o h_{j2} + \alpha w_{j2}^o(t) - \alpha w_{j2}^o(t-1),\\
&\quad \eta\delta^o h_{j3} + \alpha w_{j3}^o(t) - \alpha w_{j1}^o(t-1)).
\end{aligned}$$

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

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

Finally, the FBPN can be applied to predict the cycle time of a new lot. When a new lot is released into the fab,

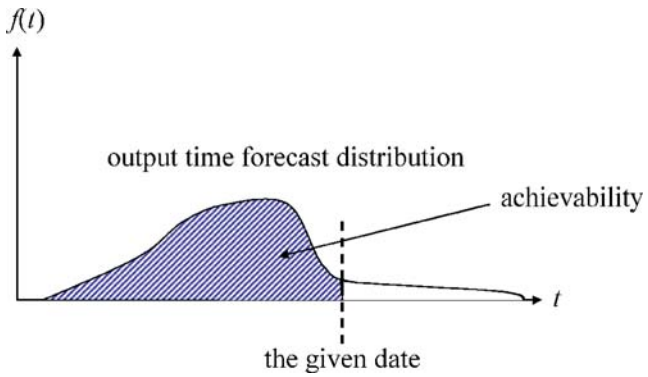

Fig. 3 The concept of achievability from a probabilistic viewpoint

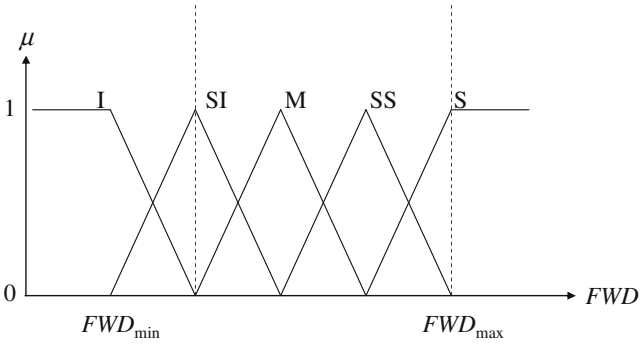

Fig. 4 The fuzzy classification of the forwardness

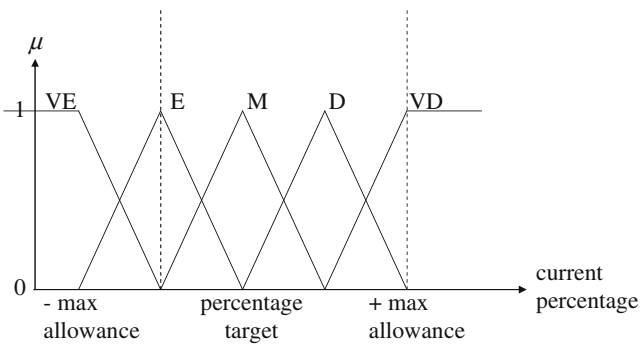

Fig. 5 Ease of priority elevation

the nine parameters associated with the new lot are recorded and compared with those of each category center. Then the FBPN with the parameters of the nearest category center is applied to forecasting the cycle time of the new lot. In this study, SOM was implemented on the software "NeuroSolutions 4.0", while a VB.NET program has been developed to implement FBPN.

3.2 Output time forecast achievability evaluation with FIR

The "achievability" of an output time forecast is defined as the possibility that the fabrication on the wafer lot can be finished in time before the output time forecast. Theoretically, if a probability distribution can be obtained for the output time forecast, then the achievability can be evaluated with the cumulative probability of the probability distribution before the given date (see Fig. 3). However, there are many managerial actions (e.g. elevating the priority of the wafer lot, lowering the priority of another wafer lot, inserting emergency lots, adding allowance, etc.) that are more influential to the achievability. Considering their effects, the evaluation of the achievability is decomposed into the following two assessments: the possible forwardness of the output time forecast if the priority is elevated, and the ease of priority elevation. For combining the two assessments, the fuzzy *and* operator is applied. The philosophy is that "if the output time forecast can be significantly forwarded after priority elevation, and the required priority elevation is not difficult at all for the lot, then the achievability of the original output time forecast is undoubtedly high, because the priority of the lot can be elevated during fabrication to achieve the given date if necessary." Finally, a set of FIR is established to facilitate the application. The procedure of applying the FIR set to evaluating the achievability of an output time forecast is detailed in the following:

1. Vary the priority of the wafer lot (from the current level to every higher level), and then predict the output time of the wafer lot again with the look-ahead SOM-FBPN.
2. Calculate the forwardness (represented with FWD) of the output time (i.e. the reduction in the cycle time) after priority elevation, and then classify the result into one of the following five categories: "insignificant (I)", "somewhat insignificant (SI)", "moderate (M)", "somewhat significant (SS)", and "significant (S)". Apply Ishibuchi's simple fuzzy partition [10] in forming the five categories (see Fig. 4).
3. Request experts to evaluate the ease of priority elevation, and then classify the result into one of the following five categories: "very easy (VE)", "easy (E)", "moderate (M)", "difficult (D)", and "very difficult (VD)". Usually the percentages of lots with various priorities in a wafer fab are controlled. The ease of priority elevation is determined against such targets (see Fig. 5).
4. Apply the fuzzy and operator to combine the two assessments. For facilitating the application, a set of fuzzy inference rules has been established in Table 3 to look up the achievability of the output time forecast, which is represented with linguistic terms including "very low (VL)", "low (L)", "medium (M)", "high (H)", and "very high (VH)".

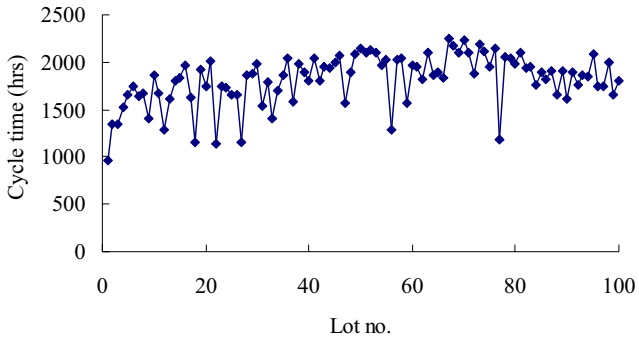

Fig. 6 Time series plot of cycle time (product A, normal lots)

Table 3 Fuzzy inference rules

Forwardness of output time forecast (%)	Ease of priority elevation	Achievability
I	–	VL
SI	VD	VL
SI	D, M, E, VE	L
M	VD	VL
M	D	L
M	M, E, VE	M
SS	VD	VL
SS	D	L
SS	M	M
SS	E, VE	H
S	VD	VL
S	D	L
S	M	M
S	E	H
S	VE	VH

Table 4 The number of wafer lot categories

	A (normal lots)	A (hot lots)	A (super hot lots)	B (normal lots)	B (hot lots)
m	6	4	4	5	3

Table 5 Comparisons of the RMSEs of various approaches

RMSE	BPN	FBPN	CBR	EFR	Look-ahead SOM-FBPN
A (normal)	178.59	177.1 (−1%)	172.44 (−3%)	164.29 (−8%)	141.47 (−21%)
A (hot)	102.1	102.27 (+0%)	86.66 (−15%)	66.21 (−35%)	59.51 (−42%)
A (super hot)	13.49	12.23 (−9%)	11.59 (−14%)	9.07 (−33%)	9.07 (−33%)
B (normal)	289.22	286.93 (−1%)	295.51 (+2%)	208.28 (−28%)	178.42 (−38%)
B (hot)	77.61	75.98 (−2%)	78.85 (+2%)	44.57 (−43%)	38.59 (−50%)

4 A demonstrative example from a simulated wafer fab

In practical situations, the history data of each lot is only partially available in the factory. Further, some information of the previous lots such as Q_n, BQ_n, and FQ_n is not easy to collect on the shop floor. Therefore, a simulation model is often built to simulate the manufacturing process of a real wafer fabrication factory [1, 3–6, 8, 11, 18]. Then, such information can be derived from the shop floor status collected from the simulation model [4]. To generate a demonstrative example, a simulation program coded using Microsoft Visual Basic 6.0 is constructed to simulate a wafer fabrication environment with the following assumptions:

1. The distributions of the interarrival times of orders are exponential.
2. The distributions of the interarrival times of machine downs are exponential.
3. The distribution of the time required to repair a machine is deterministic.
4. The percentages of lots with different product types in the fab are predetermined. As a result, this study is only focused on fixed-product-mix cases.
5. The percentages of lots with different priorities released into the fab are controlled.
6. The priority of a lot cannot be changed during fabrication.
7. Lots are sequenced on each machine first by their priorities, then by the first-in-first-out (FIFO) policy. Such a sequencing policy is a common practice in many foundry fabs.
8. A lot has equal chances to be processed on each alternative machine/head available at a step.
9. A lot cannot proceed to the next step until the fabrication on its every wafer has been finished.
10. No preemption is allowed.

The basic configuration of the simulated wafer fab is the same as a real-world wafer fabrication factory which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. Assumptions (1)∈(3), and (7)∈(9) are commonly adopted in related researches (e.g. [3–6]), while assumptions (4)∈(6) are made to simplify the situation. There are five products (labeled as A∈E) in the simulated fab. A fixed product mix is assumed. The percentages of these products in the fab's product mix are assumed to be 35, 24, 17, 15, and 9%, respectively. The simulated fab has a monthly capacity of 20,000 pieces of wafers and is expected to be fully utilized (utilization=100%). POs with normally distributed sizes (mean=300 wafers; standard deviation=50 wafers) arrive according to a Poisson process, and then the corresponding MOs are released for these POs a fixed time after. Based on these assumptions, the mean inter-release time of MOs into the fab can be obtained as (30.5*24)/(20,000/300)=11 h. An MO is split into lots of a standard size of 24 wafers per lot. Lots of the same MO are released one by one every 11/(300/24)=0.85 h. Three types of priorities (normal lot, hot lot, and super hot lot) are randomly assigned to lots. The percentages of lots with these priorities released into the fab are restricted to be approximately 60, 30, and 10%, respectively. Each product has 150∈200 steps and 6∈9 reentrances to the most bottleneck machine. The singular production characteristic "reentry" of the semiconductor industry is clearly reflected in the example. It also shows the difficulty for the production planning and scheduling people to provide an accurate due-date for the product with such a complicated routing. Totally 102 machines (including alternative machines) are provided to process single-wafer or batch operations in the fab. Thirty replicates of the simulation are successively run. The time required for each simulation replicate is about 15 min on a PC with 256 MB RAM and Athlon 64 Processor 3000+CPU. A horizon of 24 months is simulated. The maximal cycle time is less than 3 months. Therefore, 4 months and an initial WIP status (obtained from a pilot simulation run) seemed to be sufficient to drive the simulation into a steady state. The statistical data were collected starting at the end of the fourth month. For each replicate, data of 30 lots are collected and classified by their product types and

Table 6 The k values for different product types and priorities

	A (normal lots)	A (hot lots)	A (super hot lots)	B (normal lots)	B (hot lots)
k	8	6	4	9	5

Table 7 Some results of output time forecast achievability evaluation

Lot number	Priority elevation	Forwardness of output time forecast (assessment)	Ease of priority elevation	Achievability
P034	Normal→hot	−11.8% (SI)	D	L
P034	Normal→super hot	−20% (M)	VD	VL
P195	Hot→super hot	−9.6% (SI)	D	L
P026	Super hot	–	VD	–

priorities. In total, data of 900 lots can be collected as training and testing examples. Among them, 2/3 (600 lots, including all product types and priorities) are used to train the network, and the other 1/3 (300 lots) are reserved for testing. The three parameters in calculating the future discounted workloads are specified as: T_1=1 week; T_2=1.5 weeks; T_3= 2 weeks.

The time series plot of 100 simulated cycle time data is shown in Fig. 6. As we can observe here, the pattern of the cycle time is not stable and very non-stationary. The traditional approach by human decision is very inaccurate and very prone to failure when the shop status is totally different even for the same product. A trace report was generated every simulation run for verifying the simulation model. The simulated average cycle times have also been compared with the actual values to validate the simulation model.

5 Results and discussions

The first part of the hybrid system is a look-ahead SOM-FBPN applied to predicting the output time for every lot in the wafer fab. In the demonstrative example, the look-ahead SOM-FBPN and four other approaches (BPN, FBPN, CBR, and EFR) were all applied for comparison to five test cases containing the data of full-size (24 wafers per lot) lots with different product types and priorities.

In the BPN or FBPN, there is one hidden layer with six nodes. In the look-ahead SOM-FBPN, firstly wafer lots are classified with SOM. After the training and post-processing of SOM, the number of wafer lot categories (m) is determined for each product type and priority (see Table 4). Subsequently, examples of different categories are then learned with different FBPNs but with the same topology. The convergence condition was established as either the improvement in the RMSE becomes less than 0.001 with one more epoch, or 1,000 epochs have already been run.

The minimal RMSEs achieved by applying the five approaches to different cases were recorded and compared in Table 5. As noted in Chang and Liao [5], the *k*-nearest-neighbors based CBR approach should be compared with a BPN trained with only randomly chosen *k* cases. The latter was also adopted as the comparison basis, and the percentage of improvement on the minimal RMSE by applying another approach is enclosed in parentheses following the performance measure. The optimal value of parameter *k* in the CBR approach was equal to the value that minimized the RMSE [5] (see Fig. 5). The *k* values for different product types and priorities are summarized in Table 6. According to experimental results, the following discussions are made:

1. From the effectiveness viewpoint, the prediction accuracy (measured with the RMSE) of the look-ahead SOM-FBPN was significantly better than those of the other approaches by achieving a 21∈50% (and an average of 37%) reduction in the RMSE over the comparison basis-the BPN. The average advantages over CBR and EFR were 31 and 8%, respectively.
2. In the case that the lot priority was the highest (super hot lot), the look-ahead SOM-FBPN has the greatest advantage over BPN and FBPN in forecasting accuracy. In fact, the cycle time variation of super hot lots is

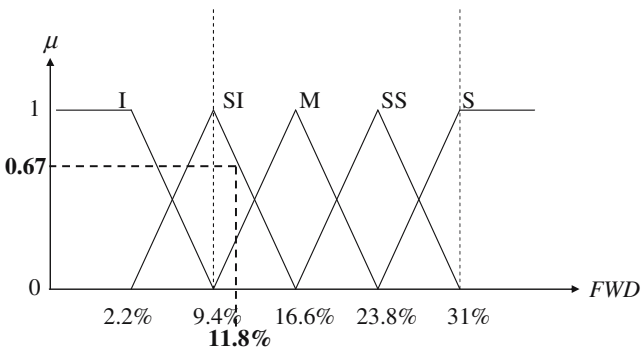

Fig. 7 The forwardness assessment result

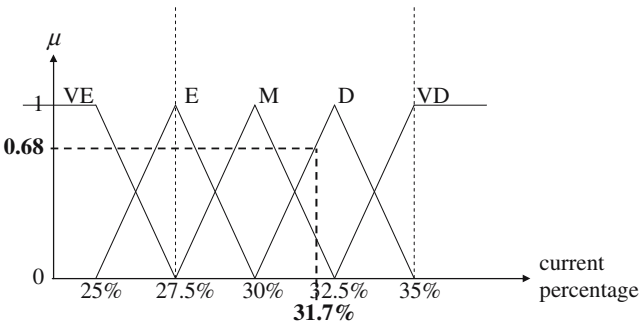

Fig. 8 The ease of priority elevation assessment result

the smallest, which makes their cycle times easy to predict. Clustering such lots seems to provide the most significant effect on the performance of cycle time prediction.
3. As the lot priority increases, the superiority of the look-ahead SOM-FBPN over BPN and FBPN becomes more evident.
4. The greatest superiority of the look-ahead SOM-FBPN over EFR happens when the lot priority is the smallest (normal lots).
5. The differences in the efficiencies of the five approaches were not significant.

The second part of the hybrid system is a set of FIR applied to evaluating the achievability of an output time forecast. Some results are shown in Table 7. Take lot P034 as an example. After elevating its priority from "normal lot" to "hot lot", the percentage of the forwardness of the output time forecast is 11.8%, which is classified as "moderate (M)" to which the highest membership function value belongs (see Fig. 7). The percentage of "hot lots" in the fab is controlled to be about 30%. At the time lot P034 is released, the percentage of "hot lots" in the fab is 31.7%. According to Fig. 8, the ease of priority elevation is evaluated as "difficult (D)". After looking up the FIR table, the achievability of the original output time forecast is "low (L)".

6 Conclusions and directions for future research

A hybrid look-ahead SOM-FBPN and FIR system is constructed in this study for lot output time prediction and achievability evaluation in a wafer fab. In the first part of the hybrid system, a look-ahead SOM-FBPN is proposed to predict the output time of a wafer lot. In the second part, a set of fuzzy inference rules is established to evaluate the achievability of an output time forecast, which is defined as the possibility that the fabrication on the wafer lot can be finished in time before the output time forecast. Achievability has been ignored in traditional studies in this field; nevertheless, it might be much more important than accuracy and efficiency from a managerial and practical viewpoint. With the proposed methodology, both output time prediction and achievability evaluation can be concurrently accomplished. For demonstrating the applicability of the proposed methodology, production simulation is also applied in this study to generate a demonstrative example. According to the results of experiments and subsequent analyses, the proposed methodology has the following advantages:

1. From the effectiveness viewpoint, the prediction accuracy of the proposed look-ahead SOM-FBPN was significantly better than those of many traditional approaches
2. The concept of achievability has not been discussed in traditional approaches, but the hybrid look-ahead SOM-FBPN and FIR system was able to evaluate the achievability

Conversely, there are also disadvantages associated with the proposed methodology:

1. The way of incorporating the fab's future release plan in the proposed methodology is subjective. For the same purpose, there are many other possible ways that can be tried to achieve better performance.
2. Compared with some traditional methods (e.g. BPN and CBR), more data are required with the proposed methodology for the sake of incorporating the fab's future release plan and classifying wafer lots.

The main contribution to the body of the knowledge is:

1. Classifying wafer lots and incorporating the fab's future release plan are shown to be both good ways of getting better performance in predicting the output time of a wafer lot.
2. A systematic procedure is proposed to embody the concept of the "achievability" of an output time forecast, which is the third performance measure in addition to the traditional accuracy and efficiency. In fact, maximizing the output time forecast achievability for each lot in the wafer fab leads to the minimization of the number of tardy jobs for the wafer fab, which is a common and very important goal of job sequencing and scheduling to the wafer fab.

However, to further evaluate the advantages and disadvantages of the proposed methodology, it has to be applied to a full-scale actual wafer fab. In addition, the proposed methodology can also be applied to cases with changing product mixes or loosely controlled priority combinations, under which the cycle time variation is often very large. These constitute some directions for future research.

References

1. Barman S (1998) The impact of priority rule combinations on lateness and tardiness. IIE Trans 30:495–504
2. Chandiramani V, Jayaseelan R, Nathan VSL, Priya KS (2004) A neural network approach to process assignment in multiprocessor systems based on the execution time. Proceedings of International Conference on Intelligent Sensing and Information Processing (ICISIP 2004), Chennai, India, January 2004, pp 332–335
3. Chang P-C, Hsieh J-C (2003) A neural networks approach for due-date assignment in a wafer fabrication factory. Int J Ind Eng 10(1):55–61
4. Chang P-C, Hsieh J-C, Liao TW (2001) A case-based reasoning approach for due date assignment in a wafer fabrication factory. Proceedings of the International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, BC, July 2001

5. Chang P-C, Hsieh J-C, Liao TW (2005) Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory. J Intell Manuf 16:549–557
6. Chen T (2003) A fuzzy back propagation network for output time prediction in a wafer fab. J Appl Soft Comput 2/3F:211–222
7. Chiang J-H (1998) A hybrid neural network model in handwritten word recognition. Neural Netw 11:337–346
8. Chung S-H, Yang M-H, Cheng C-M (1997) The design of due date assignment model and the determination of flow time control parameters for the wafer fabrication factories. IEEE Trans Compon Packaging Manuf Technol, Part C 20(4):278–287
9. Foster WR, Gollopy F, Ungar LH (1992) Neural network forecasting of short, noisy time series. Comput Chem Eng 16(4):293–297
10. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA
11. Hung Y-F, Chang C-B (2001) Dispatching rules using flow time predictions for semiconductor wafer fabrications. Proceedings of the 5th Annual International Conference on Industrial Engineering Theory, Applications and Practice, Taiwan, December /2000
12. Jiang Y, Zhou ZH (2004) SOM ensemble-based image segmentation. Neural Process Lett 20:171–178
13. Lin C-Y (1996) Shop floor scheduling of semiconductor wafer fabrication using real-time feedback control and prediction. PhD Thesis, Engineering-Industrial Engineering and Operations Research, University of California at Berkeley
14. Little JDC (1961) A proof of the queuing formula L=λ W. Oper Res 9:383–387
15. Piramuthu S (1991) Theory and methodology: financial credit-risk evaluation with neural and neural fuzzy systems. Eur J Oper Res 112:310–321
16. Ragatz GL, Mabert VA (1984) A simulation analysis of due date assignment. J Oper Manag 5:27–39
17. Tiwari MK, Roy D (2002) Minimization of internal shrinkage in casting using synthesis of neural network. Int J Smart Eng Syst Des 4:205–214
18. Vig MM, Dooley KJ (1991) Dynamic rules for due-date assignment. Int J Prod Res 29(7):1361–1377
19. Wang L-X, Mendel JM (1992) Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 22(6): 1414–1427
20. Weeks JK (1979) A simulation study of predictable due-dates. Manage Sci 25:363–373