J Intell Manuf (2009) 20:421–429
DOI 10.1007/s10845-008-0116-2

Evolving CBR and data segmentation by SOM for flow time prediction in semiconductor manufacturing factory

Pei-Chann Chang · Chin Yuan Fan · Yen-Wen Wang

Received: 18 September 2007 / Accepted: 13 May 2008 / Published online: 31 May 2008

Abstract Flow time of semiconductor manufacturing factory is highly related to the shop floor status; however, the processes are highly complicated and involve more than 100 production steps. Therefore, a simulation model with the production process of a real wafer fab located in Hsin-Chu Science-based Park of Taiwan is built for further studying of the relationship between the flow time and the various input variables. In this research, a hybrid approach by combining Self-Organizing Map (SOM) and Case-Based Reasoning (CBR) for flow time prediction in semiconductor manufacturing factory is developed. And Genetic Algorithm (GA) is applied to fine-tune the weights of features in the CBR model. The flow time and related shop floor status are collected and fed into the SOM for clustering. Then, a corresponding SGA-CBR method is selected and applied for flow time prediction. Finally, using the simulated data, the effectiveness of the proposed method (SGA-CBR) is shown by comparing with other approaches.

Keywords Due-date assignment · Flow time prediction · Case-based reasoning · Genetic algorithms · Self-organizing map

P.-C. Chang (✉)
Department of Information Management, Yuan Ze University, No. 135, Yuan-Tung Rd., Chung-li, Tao-Yuan 32026, Taiwan, R.O.C.
e-mail: iepchang@saturn.yzu.edu.tw

C. Y. Fan
Department of Industrial Engineering & Management, Yuan Ze University, No. 135, Yuan-Tung Rd., Chung-li, Tao-Yuan 32026, Taiwan, R.O.C.

Y.-W. Wang
Department of Industrial Engineering & Management, Ching-Yun University, Tao-Yuan 32026, Taiwan, R.O.C.

Introduction

The flow time of a wafer lot is the total processing time that the wafer lot is processed through the whole production cycle, and is equal to the sum of the production lead-time. Predicting the flow time for every lot in a wafer fab is a critical task not only to the fab itself, but also to its customers. That is only after the flow time of each lot in a wafer fab is accurately predicted; several managerial goals including internal due-date assignment, customer due-date confirmation, order release to each manufacturing process, and ordering decision support then can be simultaneously achieved.

Traditionally, assigning due date for each order is accomplished by the production planning and control staffs based on their knowledge of the manufacturing processes and shop floor status. The production planning and scheduling staffs usually estimate the flow time of each order based on products manufactured before and schedule its release to the shop floor for production. Even if the product specification is exactly the same, the status of the shop floor such as jobs in the system, shop loading and jobs in the bottleneck machine may not be identical to the previous production. As a result, due date estimated by the production planning and scheduling staffs might be subject to errors.

As the advance in artificial intelligence (AI), tools in soft computing have been widely applied in manufacturing planning and scheduling problems. Chang and Hsieh (2003) reported that back-propagation neural networks (BPN) could be more effective than some traditional direct procedures for due date assignment since neural network can obtain a probable result even if the input data are incomplete or noisy. Using a k-nearest-neighbors (KNN) based case-based reasoning (CBR) approach with dynamic feature weights and non-linear similarity functions; Chiu et al. (2003) found that further performance improvement could be made.

As Sfetsos and Siriopoulos (2004) mentioned, the kind of industrial applied forecasting researches can be classified in two categories: the hybrid schemes and combinatorial synthesis of multiple models. Hybrid methodologies are usually formed by adding a clustering scheme with general forecasting model. The clustering algorithm distinguishes smaller groups of data that have similar characteristics, and then the forecasting schemes are developed within each group. Successful applications of hybrid forecasting schemes can be found by Chang et al. (2001); Cao (2003), Lin and Xu (2006). Combinatorial schemes are constructed by the means of combining individual models to find the better forecasting models. Generally, neural network models, fuzzy logic, time-series functions, and various evolving algorithms were widely applied while the combinatorial scheme is developed (Chiang et al. 2006; Kodogiannis and Lolis 2002; Carpinteiro et al. 2007; Koutroumanidis et al. 2006; Kalaitzakis et al. 2002; Milidiú et al. 1999; Chen and Burrell 2001; Watson and Gardingen 1999).

This paper constructs a case-based prediction system with the aid of a Self-Organizing Map (SOM), Genetic Algorithm (GA) and CBR, and we call it SGA-CBR in the rest of the article. The SOM is first used to classify the data, and after the classification GA is applied to construct the CBR prediction method by searching for the best weight combination.

The rest of the paper is organized as follows: Sect. "Literature survey" reviews some related literatures. Section "Problem statements" briefly describes the case that will be discussed in this research. Section "A hybrid system combining SOM and GA-CBR" presents the framework of the methodology applied in the flow time prediction method. Section "Experimental results" presents some experimental results of various models including other compared methods. Section "Conclusions" discusses the simulated results from these different models and then the conclusion is made.

Literature survey

Background of due-date assignment

Song et al. (2002) have mentioned two principles in due-date assignment problem are based upon analytical and empirical methods. Analytic methods include queuing networks and linear or non-linear programming. The due date is assigned by minimizing a cost function. Bookbinder and Noor (1985) consider a single machine due date assignment problem with a customer service level constraint. Job processing times are modeled as independent exponentially distributed random variables. Philipoom et al. (1997) use linear programming and neural networks for setting due dates by minimizing an asymmetric earliness and tardiness cost function. The analytic methods above are applied to job shops with stochastic processing times. However, these do not include assembly processes.

Network methods such as the critical path method (CPM) and the program evaluation and review technique (PERT) have been widely used for planning projects in the design and manufacturing of capital goods. The effect of interactions of different paths on the completion time is studies by Sculli (1983); Anklesaria and Drezner (1986); Pontrandolfo (2000) and Elmaghraby et al. (2000). Yano (1987) considers stochastic lead-time in a simple two level assembly system, with different processing time distributions including Poisson and Negative binomial. Song et al. (2002) apply a two-stage network to estimate the assembly activity at each assembly stage to assign product due dates for complex products with stochastic manufacturing and assembly processing times.

The second type of due date assignment problem is empirical methods which are based on production planning and scheduling staffs' intuitions or working experiences to estimate the flow-time. If the product specification is exactly the same, a flow time can be derived and the due-date of the product is assigned. Understandably, the status of the shop floor such as jobs in the system, shop loading and jobs in the bottleneck machine may not be all the same. As a result, the due date estimated could be subject to errors. Cheng and Gupta (1985) propose several different due-date assignment approaches: the TWK (total processing time), NOP (number of operations), CON (constant allowance), and RDM (random allowance) rules. As soon as the processing times are estimated by these rules, the due date is set equal to the order release time plus the estimated processing time, i.e., $d_i = r_i + p_i$ where d_i is the due-date of the i-th order, r_i and p_i are the release time and processing time of order i respectively.

Many other discussions focus on the relationships between the shop status information and due-dates. Several significant effective factors, for example, jobs-in-queue (JIQ), jobs-in-system (JIS), delay-in-queue (DIQ), and processing plus waiting times (PPW) were explored. Conway et al. (1967) revealed that due-date rules incorporating job characteristics performed better than those ignoring job characteristics.

Approaches in due-date assignment

In recent years, many artificial intelligent and soft computing methods have been used for decision support and forecasting. Three data mining algorithms including clustering, KNN and Regression Tree are used to develop nonlinear predictors applicable to the majority of process lots Backus et al. (2006) and they are compared with respect to performance in actual manufacturing data (to predict times for both final and intermediate steps) and for the feasibility to maintain and rebuild the model.

As mentioned and Chen (2007a), 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 (Multiple Factor Linear Combination): the cycle time of a lot is estimated with the weighted sum of parameters including the following:
   (a) Job properties: the total processing time, the number of re-entrances, and the number of operations of the lot.
   (b) Cycle time and waiting time series: the actual cycle times, the waiting times, the total processing times, the numbers of re-entrances, and the numbers of operations of some (usually three) most recently completed lots.
   (c) Workload information: the number of jobs (work-in-progress, WIP) in the fab or waiting for the most bottleneck machines or on the processing route of the lot, the average fab utilization.

   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 (Production Simulation): a fab production simulation system continuingly updating the related databases to maintain enough validity can also be applied to predicting/simulating the output time of a wafer lot (e.g. Ragatz and Mabert 1984; Weeks 1979). 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. inconsistent human-assisted operations, unexpected machine downs, etc.), so as to obtain a more reliable forecast. There are two shortages of PS: (i) huge amount of data need to be maintained; (ii) simulation time is often lengthy. Nevertheless, PS is the most accurate output time prediction approach (if the related databases are continuing 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 Chang and Lai (2005).
(3) BPN (Back Propagation Networks): many studies have shown that artificial neural networks (ANN) outperform traditional methods in time series forecasting by Foster et al. (1992). The advantages of a BPN include the tolerance of noises Piramuthu (1991), the speed of application, and the capability of simulating complex systems (such as a wafer fab). Chang et al. (2005) and Chang and Hsieh (2003) both forecast 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 square errors (RMSE) is 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 (Case-Based Reasoning): Chang et al. (2001) and Chang and Lai (2005) propose a KNN 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. The CBR approach outperforms the BPN approach in forecasting accuracy.
(5) Fuzzy modeling methods: Chang et al. (2005) and Chang and Liao (2006) modify the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the WM method with a simple GA and propose the Evolving Fuzzy Rule (EFR) approach to predict the cycle time of a wafer lot. Their EFR approach outperforms CBR and BPN in prediction accuracy. Genetic techniques have shown to be capable of carrying out a comprehensive optimization of the parameters. Chang et al. (2008) present a Neural-Fuzzy model for the flow time estimation using simulated data generated from a Foundry Service company. This Neural-Fuzzy model applies influential factors identified from the shop floor to estimate the flow time of a new order. The fuzzy neural network is trained using a back-propagation algorithm to adjust the weight coefficients of the network and the parameters of the fuzzy membership functions. The trained network performs better than the CBR method and a Multi-Layer Perceptrons Neural Network.
(6) Hybrid approaches: Chen (2006, 2007b,c) construct a FBPN that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN is a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN in the efficiency respect. In the respect of prediction accuracy measured with the minimal RMSE, the performance of the FBPN is slightly better than that of the BPN.

To the best of our knowledge, none of the above studies considered the non-linear feature value distance between an old case and a new case. Therefore, this paper aims to investigate the effect of GA-based feature weighting together with a number of non-linear similarity functions.

Problem statements

The basic configuration of the wafer fabrication factory is the same as a real-world one located in the Science-Based Park in Hsin-Chu, Taiwan, R.O.C. There are 66 single-server

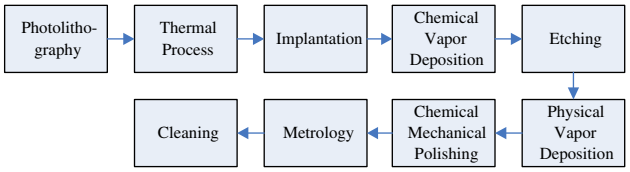

Fig. 1 Basic front-end processes

Fig. 2 Basic back-end processes

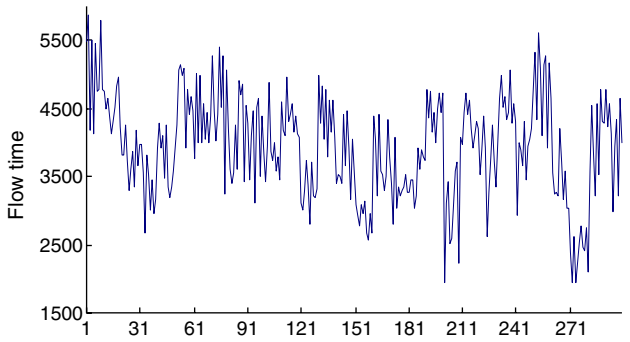

Fig. 3 Time series plot of flow time

or multiple-server workstations in the shop floor. The major wafer manufacturing processes are divided into two sections, i.e., the front-end process and the back-end process. A flowchart of the basic front-end processes is described in Fig. 1. The production steps are just a step-by-step process. Real floor shop manufacturing processes are more complicated with many detailed processing procedures. After the front-end processes, wafers are fed into the back-end processes. A simple flowchart of the back-end processes is also shown in Fig. 2.The time series plot of 300 flow time data is depicted in Fig. 3 and the pattern of the flow time is not stable in this plot as the data fluctuates up and down abruptly. The traditional approach by human decision is very inaccurate and very prone to fail when the shop status is totally different even for the same product. This is the motive for this research to develop an approach to reduce the forecasting error based on such a non-stationary situation.

A hybrid system combining SOM and GA-CBR

This research first uses a self-organization map (SOM) to cluster past cases into different groups, and the training cases in each sub-group are used to train the best weight of each feature by GA. During the testing process, the most similar sub-group to the new case will be retrieved by CBR from case base. A new case is compared to each case within the selected group in order to find the most similar case to derive the forecasting flow time of the new case. The framework of SGA-CBR is described in Fig. 4. Three soft computing techniques are applied in this research and they include a SOM, a GA and a CBR tool. SOM is applied to cluster the historic data; GA is employed to evolve the feature's weight of a CBR system. The system is tested on a simulated data from a fab located in Taiwan and there are totally 300 records of data that are randomly divided into 240 records of training data and 60 records of testing data using a 4-fold testing procedure.

Basically, there are three major processes in SGA-CBR model: a data pre-processing by SOM, system training and system testing.

Data pre-processing by SOM

Step 1: Classify the training data by SOM

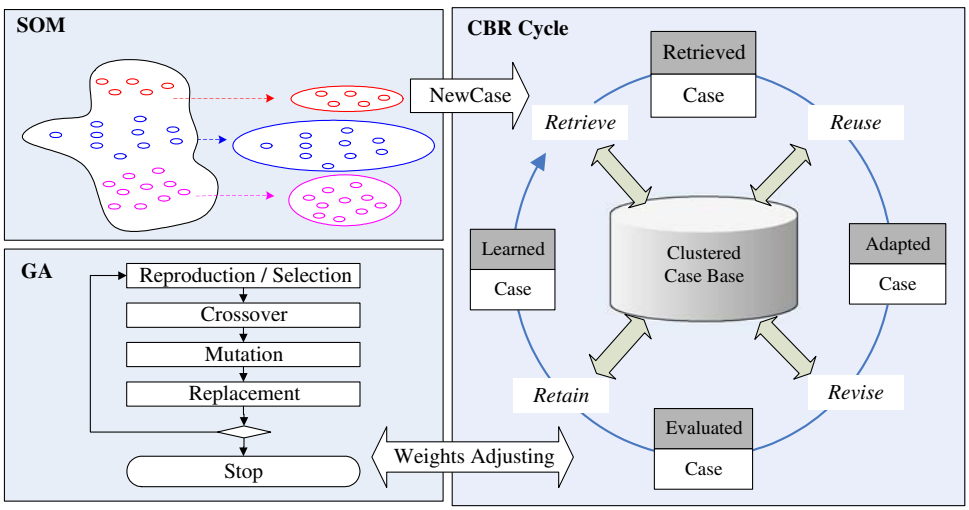

Fig. 4 The framework of SGA-CBR model

In the historic orders, each data collected from the shop floor will include a flow time which has occurred, i.e., Y, and a set of features including order quantities (X_1), existing order qualities (X_2), average shop workload (X_3), average queue length (X_4), the queue length of bottleneck workstation (X_5), and utilization rate of work station (X_6), when a new order arrives. These six features of a new order will be the input variables to SOM, and SOM will produce output processing elements similar to neighboring elements. The cases in the same group would have similar connection weight. Detailed procedures of the SOM model are described in Appendix A.

After the set of data has been processed by SOM, a new case (a new order) can be categorized into a pre-defined group.

System training

Step 2: Initial weights generation

Randomly generate the initial weights W_j^i of the j-th feature in sub-group i.

Step 3: Case retrieving

The system will retrieve the most similar cases from case base using similarity rule in order to predict the flow time for a new case. There are two cases in group i, i.e., C_m^i and C_n^i. The similarity coefficient of these two cases, i.e., S_{mn}, is calculated as follows:

$$S_{mn} = Dis\left(C_m^i, C_n^i\right), \quad \forall n \neq m \tag{1}$$

$Dis()$ is the distance between two cases and $Dis()$ is computed as follows:

$$Dis\left(C_m^i, C_n^i\right) = \sqrt{\sum_f W_j^i \left(F_f^m - F_f^n\right)^2} \tag{2}$$

where F_f^m means the value of the f-th feature of case m. Thus, $Dis(C_m^i, C_n^i)$ computes the summarized weighted distance between case m and n.

Step 4: Case reusing

For the set of similar cases retried from the last step, a KNN procedure is employed to gain more matching cases to forecast the flow time of the new case. For example, when $k = 5$ in the sub-group, the forecasted flow time of new case is determined by the average flow time of these 5 best matching cases. The parameter k of each sub-group is determined by trail-and-error. Normally, 5 neighbors will give the best forecasting results.

Step 5: Error computing

Root of mean square error (RMSE) is adopted to be the performance measure in this research.

$$RMSE = \sqrt{\frac{\sum_{l=1}^{N} (forecasted\ value - real\ value)^2}{N}} \tag{3}$$

where, N is the total number of cases in the sub-group.

Step 6: Weights adjusting by GA

CBR emphasizes on how to describe and retrieve cases, and one of the crucial points is the combination of the weight for each characteristic factor. In this research, GA is applied to evolve the weight of each feature and a near optimal feature weight will be generated for each sub-group. To properly setup the parameters in GA, Taguchi experiment design is applied in fine-tuning the parameters such as the number of populations, crossover rate, mutation rate, selection method, crossover method, mutation method etc. Parameters setting of GA are listed in following:

In addition, detailed procedures of GA approaches are listed in Appendix B.

Step 7: Cases and weight of each feature retaining

The best weight combination of each sub-group is retained for a later testing process.

System Testing

Step 8: New test case retrieving

The same as process above, similarity rule is used to compute the similarity of cases.

Step 9: New testing case reusing

Find the most k similar cases of new case.

Step 10: Forecasted flow time generating

Forecast the flow time of new case from k similar cases.

Experimental results

Data clustered by SOM

The main purpose of data clustering is to reduce the effect of data noise therefore a more homogenous data set can be found within each cluster. SOM is applied in this research to cluster the historic data. The final clustering results by SOM can be found in Fig. 5, which shows the results of two and three clusters for these 240 historic simulated data.

It is interesting to observe that the data within each cluster may be different. As the number of clusters increase, the number of data within each cluster will decrease. It should be avoided when the number of clusters is too large, e.g. 10, and then there are too small numbers of data within each cluster. The SGA-CBR model may not be able to come out with good forecasting results since the model is not general enough to cover all different cases. It is expected that the number of clusters might influence the forecasting result. Therefore, the number of clusters will be further discussed in the next sub-section.

SGA-CBR in different clusters

The forecasted results of SGA-CBR model under different number of clusters are shown in Fig. 6 and the parameters

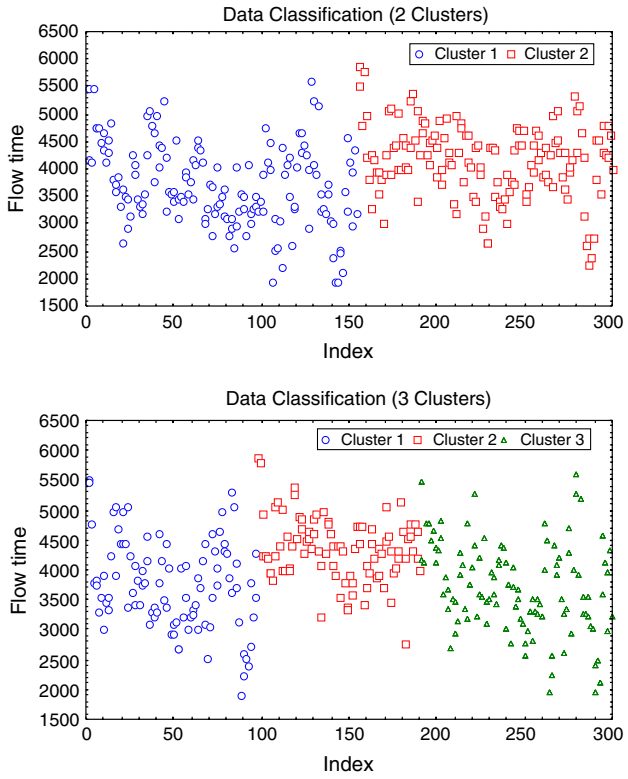

Fig. 5 The two and three clustered results by using SOM

setting in GA are shown in Table 1. According to Fig. 6, when the number of clusters is increasing, the forecasted data will be more fitted into the real data. Part of the reasons is because the data are more homogeneous within these clusters and the predicator generated using these data in the cluster are more representative. Thus the overall performance of the SGA-CBR is greatly improved. This phenomenon is also observed in Fig. 7, i.e., Mean Absolute Percentage Error (MAPE) and RMSE performances of SGA-CBR model in different number of clusters. The errors of the forecasted

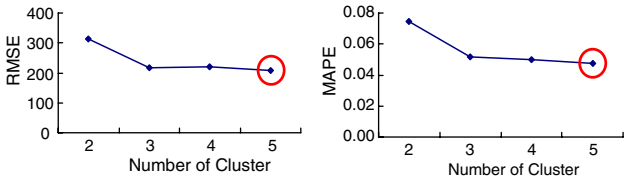

Fig. 7 MAPE and RMSE performances of SGA-CBR in different number of clusters

Table 1 Parameters settings in GA

Parameters	Setting
Selection method	Binary tournament
Crossover method	Single point crossover
Crossover rate	0.85
Mutation method	Swap mutation
Mutation rate	0.1
Reproduction	Elitism strategy
Population size	30
Stopping criteria	1,000

results quickly decreased and converged when the number of clusters increased from 2 to 5. It shows that the data segmentation does take effect and the forecasting power of the clustered data is increasing. However the optimal decision of the number of clusters is still open to the academics since there are no exact theorems proved in explaining this phenomenon.

According to Fig. 7, the data segmented in 5 clusters provides the best forecasting accuracy and it is selected as the best number of sub-groups for SGA-CBR in the research. As shown in Table 2, i.e., the performances of SGA-CBR in MAPE and RMSE under different number of clusters are listed. Once again, it is observed that when the number of clusters is larger than 3, the accuracy of the proposed model will converge. There is no obvious improvement when the number of clusters is further increased. Actually, when the

Fig. 6 SGA-CBR with different number of cluster

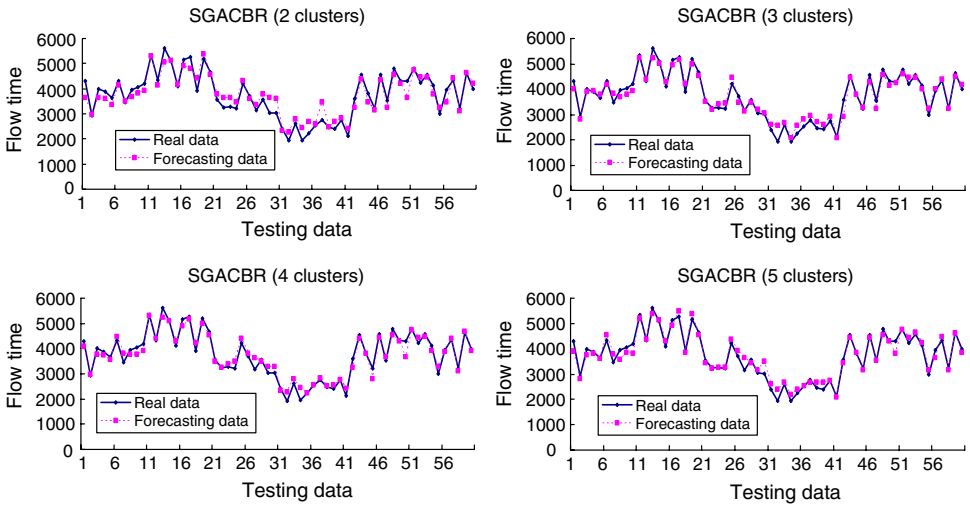

Table 2 The performances of SGA-CBR in MAPE and RMSE under different number of clusters

Number of cluster	2	3	4	5
MAPE (%)	7.44	5.18	5.01	4.73
RMSE	312.5231	218.3532	218.6860	208.2776

number of clusters reaches certain limit, e.g. 7, the accuracy will drop instead. The reason is that the data sample in each cluster is too small therefore the generality of the forecasting model generated from each cluster is decreasing. Therefore, the number of clusters will be 5 after experimental tests.

Comparison with other methodology

Other forecasting methodologies are compared with SGA-CBR in this research, such as general CBR, BPN, GA and fuzzy rule based method (GA&WM), GA and CBR hybrid method (GA-CBR), and Fuzzy rule based SOM method (SOM&WM). The detail of these methods please refers to the previous researches (Chang et al. 2001; Chang and Hsieh 2003; Chang and Lai 2005; Chang et al. 2005; Chang and Liao 2006).

By observing Table 3, SGA-CBR proposed in this research performs superior to other methods that performed well in previous research. The reason why SGA-CBR of this research outperforms others is because GA can fine-tune the weights. CBR is one of the famous forecasting methods while resolving this kind of forecasting problem with multiple features considering. By adopting the Euclidean distance to retrieve the similar cases, CBR is an effective and efficient method. Otherwise, in the real world, each feature may play a different important role. It means we should take different importance of each feature into consideration; thus, we use GA to search the best weights combination of features in our CBR process.

In the comparative study, the overall average RMSE of SGA-CBR is 208, the overall average RMSE of other methods can be found in Table 3. Hence the results of our limited comparative studies show that the proposed SGA-CBR method produces the lowest RMSE value.

Table 3 Performance comparisons of different forecasting models

Methodology	RMSE	Improving rate (%)
CBR	538	–
BPN	480	10.78
GA&WM	479	10.97
GA-CBR	391	27.32
SOM&WM	320	40.52
SGA-CBR	208	61.34

Conclusions

In this paper, a novel SGA-CBR model is presented to help semiconductor manufacturing companies in forecasting the due date of a new coming order. Semiconductor manufacturing companies can follow this due date to make plans and to coordinate related production activities thereafter.

The contributions of this research include the following:

1. An SOM neural network is employed to cluster the historic data into sub-clusters. Then a new case, i.e., a new coming order, can be categorized into a pre-defined cluster.
2. A SGA-CBR model is then generated using data within the selected cluster with a more homogeneous data set. The experimental results demonstrate the effectiveness of the SGA-CBR model that is superior to other earlier approaches.

This proposed approach in flow time prediction might be interested to other academic researchers and industrial engineers and managers:

Data noise and high dimensionality are two notorious problems in flow time prediction or time series data processing. These two problems will influence the forecasting accuracy significantly however there seems to be no effective methods in handling these issues in traditional approaches. As observed in recent researches, data preprocessing seems to be very important in selecting significant variables and clustering the data into more homogeneous classes. It is a very interesting subject to be further investigated since there is no exact theorem explaining the effect of number of clusters to the forecasting accuracy. In addition, a novel hybrid model by combining different soft-computing tools such as fuzzy sets, neural networks, and evolutionary algorithms can be further studied in the near future.

Appendix A: Detailed procedures of SOM model

The detailed procedures of SOM model are described as follows:

Step 1: Initialize each neuron's weight $w_i = [w_{i1}, w_{i2}, \ldots, w_{ij}]^T \in \Re^j$. In this research, neuron weights are initialized by drawing random samples from input dataset.

Step 2: Present an input pattern $x = [x_1, x_2, \ldots, x_j]^T \in \Re^j$. In this case, input pattern is a series of variables representing current shop floor status. Calculate the distance between pattern x, and each neuron weight w_i, and therefore, identify the winning neuron or best matching unit c such as

$$\|x - w_c\| = \min_i \{\|x - w_i\|\} \tag{4}$$

Euclidian distance is employed as the distance metric.

Step 3: Adjust the weights of winning neuron c and all neighbor units

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)] \quad (5)$$

where i is the index of the neighbor neuron and t is an integer, the discrete time coordinate. The neighborhood kernel $h_{ci}(t)$ is a function of time and the distance between neighbor neuron i and winning neuron c. $h_{ci}(t)$ defines the region of influence that the input pattern has on the SOM and consists of two parts: the neighborhood function $h(\|\|, t)$ and the learning rate function $\alpha(t)$, in Eq. 6.

$$h_{ci}(t) = h(\|r_c - r_i\|, t)\alpha(t) \quad (6)$$

where r is the location of the neuron on two-dimensional map grids. In this research we used Gaussian Neighborhood Function. The learning rate function $\alpha(t)$ is a decreasing function of time. The final form of the neighborhood kernel with Gaussian function is as follows:

$$h_{ci}(t) = \exp\left(\frac{\|r_c - r_i\|}{2\sigma^2(t)}\right)\alpha(t) \quad (7)$$

where α(t) defines the width of the kernel.

Step 4: Repeat steps 2 and 3 until the convergence criterion is satisfied.

Appendix B: Detailed procedures of GA

Detailed procedures of GA in finding the best combination of feature's weight are described as below:

Step 1: Encoding

The most common encoding method for gene is binary number used as the original calculating system by computer. Each factor influencing flow time is assigned a weight with the combination of eight binary numbers shown as in Fig. 8.

Step 2: Generate the initial population

Initial weights are randomly generated between 0 and 1 and these initial solutions form the first population. Operators of GA will be applied on these chromosomes later.

Step 3: Compute the fitness value

The original concept of fitness is "the larger the better", because solutions with larger fitness tend to propagate to the next generation. The objective function for the flow time prediction problem is to find RMSE of these testing data which is "the smaller the better." Therefore, the fitness function for a set of training cases, i.e., k, is calculated as follows:

$$fit = \sqrt{\frac{\sum_{i=1}^{k}(forecasted\ value - real\ value)^2}{k}} \quad (8)$$

Step 4: Reproduction/selection

The roulette wheel selection method is applied in this research and the selection probability of each chromosome represents the area proportion of each string on the roulette wheel. Therefore, a chromosome with larger fitness function value will have a greater probability to be selected for crossover. The probability $p(x)$ of each chromosome x will be defined below:

$$p(x) = \frac{fit(x)}{\sum fit(x)}. \quad (9)$$

Step 5: Crossover

After the parameter design, single point crossover method is applied in the research.

Step 6: Mutation

After the parameter design, swap mutation method is applied in the research.

Step 7: Elite strategy

Elite strategy is applied in this research in order to have greater probability for retaining good chromosomes into the next generation. 30 % of parent chromosomes and 70% offspring chromosomes are used in this research.

Step 8: Replacement

The new population generated by the previous steps updates the old population.

Step 9: Stopping criteria

If the number of generations equals to the maximum generation number, i.e., 1,000, then stop, otherwise go to step 3.

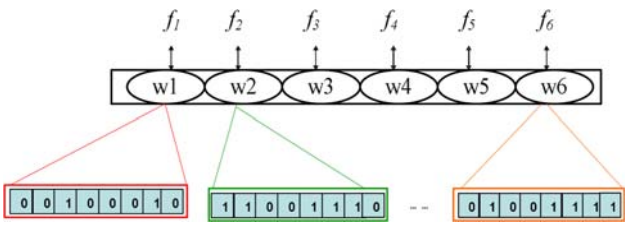

Fig. 8 Weights with combination of binary numbers

References

Anklesaria, K. P., & Drezner, Z. (1986). A multivariate approach to estimating the completion time for PERT networks. *Journal of the Operational Research Society, 37*(8), 811–815.

Backus, P., Janakiram, M., Mowzoon, S., Runger, G. C., & Bhargava, A. (2006). Factory cycle-time prediction with a data-mining approach. *IEEE Transactions on Semiconductor Manufacturing, 19*(2), 252–258.

Bookbinder, J. H., & Noor, A. I. (1985). Setting job-shop due-dates with service-constrains. *Journal of the Operational Research Society, 36*(11), 1017–1026.

Cao, L. (2003). Support vector machines experts for time series forecasting. *Neurocomputing, 51*, 321–339.

Carpinteiro, O. A. S., Lima, I., Leme, R. C, de Souza, A. C. Z., Moreira, E. M., & Pinheiro, C. A. M. (2007). A hierarchical neural model with time windows in long-term electrical load forecasting. *Neural Computing & Applications, 16*(4), 465–470

Chang, P. C., & Hsieh, J. C. (2003). A neural networks approach for due date assignment in a wafer fabrication factory. *International Journal of Industrial Engineering, 10*, 55–61.

Chang, P. C., Hsieh, J. C., & Liao, T. W. (2001). A case-based reasoning approach for due-date assignment in a wafer fabrication factory. *Lecture Notes in Computer Science, 2080*, 648–659.

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*(5), 549–557.

Chang, P. C., & Lai, C. Y. (2005). A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. *Expert Systems with Applications, 29*(1), 183–192.

Chang, P. C., & Liao, T. W. (2006). Combing SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. *Applied Soft Computing, 6*(2), 198–206.

Chang, P. C., Wang, Y. W., & Ting C. J. (2008). A fuzzy neural network for the flow time estimation in a semiconductor manufacturing factory. *International Journal of Production Research, 46*(4), 1017–1029.

Chen, T. (2006). A hybrid SOM-BPN approach to lot output time prediction in a wafer fab. *Neural Processing Letters, 24*(3), 271–288.

Chen, T. (2007a). An intelligent hybrid system for wafer lot output time prediction. *Advanced Engineering Informatics, 21*, 55–65.

Chen, T. (2007b). Predicting wafer lot output time with a hybrid FCM-FBPN approach. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, 37*(4), 784–793.

Chen, T. (2007c). A fuzzy-neural approach with BPN post-classification for job completion time prediction in a semiconductor fabrication plant. *Advances in Soft Computing, 41*, 580–589.

Chen, D., & Burrell, P. (2001) Case-based reasoning system and artificial neural networks: A review. *Neural Computing & Applications, 10*, 264–276.

Cheng, T. C., & Gupta, E. (1985). Survey of scheduling research involving due-date determination decisions. *European Journal of Operational Research, 36*(11), 1017–1026.

Chiang, C. C, Ho, M. C., & Chen, J. A. (2006). A hybrid approach of neural networks and grey modeling for adaptive electricity load forecasting. *Neural Computing & Applications, 15*(3–4), 328–338.

Chiu, C. C., Chang, P. C., & Chiu, N. H. (2003). A case-based expert support system for due date assignment in a wafer fabrication factory. *Journal of Intelligent Manufacturing, 14*, 287–296.

Conway, R. W., Maxwell, W. L., & Miller, L. W. (1967). *Theory of scheduling*. Massachusetts: Addison-Wesley.

Elmaghraby, S. E., Ferreira, A. A., & Tavares, L. V. (2000). Optimal start times under stochastic activity durations. *International Journal of Production Economics, 64*, 153–164.

Foster, W. R., Gollopy, F., & Ungar, L. H., (1992). Neural network forecasting of short, noisy time series. *Computers in Chemical Engineering, 16*(4), 293–297.

Kalaitzakis, K., Stavrakakis, G. S., & Anagnostakis, E. M. (2002). Short-term load forecasting based on artificial neural networks parallel implementation. *Electric Power Systems Research, 63*(3), 185–196.

Kodogiannis, V., & Lolis, A. (2002). Forecasting financial time series using neural network and fuzzy system-based techniques. *Neural Computing & Applications, 11*(2), 90–102.

Koutroumanidis, T., Iliadis, L., & Sylaios, G. K. (2006). Time-series modeling of fishery landings using ARIMA models and fuzzy expected intervals software. *Environmental Modelling and Software, 21*(12), 1711–1721.

Lin, C. J., & Xu, Y. J. (2006). A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications. *Fuzzy Sets and Systems, 157*(8), 1036–1056.

Milidiú, R. L., Machado, R. J., & Rentería, R. P. (1999). Time-series forecasting through wavelets transformation and a mixture of expert models. *Neurocomputing, 28*(1–3), 145–156.

Philipoom, P. R., Wiegmann, L., & Rees, L. P. (1997). Cost-based due date assignment with the use of classical and neural network approaches. *Naval Research Logistics, 44*, 21–46.

Piramuthu, S. (1991). Theory and methodology financial credit-risk evaluation with neural and euralfuzzy systems. *European Journal of Operational Research, 112*, 310–332.

Pontrandolfo, P. (2000). Project duration in stochastic networks by the PERT-Path technique. *International Journal of Project Management, 18*, 215–222.

Ragatz, G. L., & Mabert, V. A. (1984). A simulation analysis of due date assignment. *Journal of Operations Management, 5*, 27–39.

Sculli, D. (1983). The completion time of PERT networks. *Journal of the Operational Research Society, 34*(2), 155–158.

Sfetsos, A., & Siriopoulos, C. (2004). Combinatorial time series forecasting based on clustering algorithms and neural networks. *Neural Computing & Applications, 13*, 56–64.

Song, D. P., Hicks, C., & Earl, C. F. (2002). Product due date assignment for complex assemblies. *International Journal of Production Economics, 76*, 243–256.

Watson, I., & Gardingen, D. (1999). A distributed cased-based reasoning application for engineering sales support. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (Vol. 1, pp. 600–605).

Weeks, J. K. (1979). A simulation study of predictable due-dates. *Management Science, 25*, 363–373.

Yano, C. A. (1987). Stochastic lead-times in two-level assembly systems. *IIE Transactions, 19*(4), 371–378.