

Article

A Hybrid Approach Combining Fuzzy c-Means-Based Genetic Algorithm and Machine Learning for Predicting Job Cycle Times for Semiconductor Manufacturing

Gyu M. Lee ^{1,*}  and Xuehong Gao ^{1,2}¹ Department of Industrial Engineering, Pusan National University, Busan 46241, Korea; b2092579@ustb.edu.cn² Department of Safety Engineering, University of Science and Technology Beijing, Beijing 100083, China

* Correspondence: glee@pnu.edu

Abstract: Job cycle time is the cycle time of a job or the time required to complete a job. Prediction of job cycle time is a critical task for a semiconductor fabrication factory. A predictive model must forecast job cycle time to pursue sustainable development, meet customer requirements, and promote downstream operations. To effectively predict job cycle time in semiconductor fabrication factories, we propose an effective hybrid approach combining the fuzzy c-means (FCM)-based genetic algorithm (GA) and a backpropagation network (BPN) to predict job cycle time. All job records are divided into two datasets: the first dataset is for clustering and training, and the other is for testing. An FCM-based GA classification method is developed to pre-classify the first dataset of job records into several clusters. The classification results are then fed into a BPN predictor. The BPN predictor can predict the cycle time and compare it with the second dataset. Finally, we present a case study using the actual dataset obtained from a semiconductor fabrication factory to demonstrate the effectiveness and efficiency of the proposed approach.

Keywords: back-propagation network; fuzzy c-means clustering method; genetic algorithm; job cycle time prediction; semiconductor fabrication



Citation: Lee, G.M.; Gao, X. A Hybrid Approach Combining Fuzzy c-Means-Based Genetic Algorithm and Machine Learning for Predicting Job Cycle Times for Semiconductor Manufacturing. *Appl. Sci.* **2021**, *11*, 7428. <https://doi.org/10.3390/app11167428>

Academic Editor: Francesco Bianconi

Received: 22 June 2021

Accepted: 11 August 2021

Published: 12 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Job cycle time is the cycle time of a job or the time required to complete a job, the prediction of which is a critical task associated with different types of systems, such as production systems [1,2], computer systems [3], and network systems [4]. Various managerial goals can be achieved once the job cycle time has been predicted accurately at a factory, including ordering-decision support, internal due-date assignment, output projection, enhancing customer relationships, guiding subsequent operations [5], pursuing sustainable development, meeting customer requirements, and promoting downstream operations. These advantages can support the competitiveness of a system to allow it to survive and be developed sustainably.

In semiconductor manufacturing, each wafer fabrication factory is a complicated production system with idiosyncratic features such as changing demand, various product types and priorities, equipment unreliability, unbalanced capacity, job reentry into machines, alternative machines, sequence-dependent setup time, and shifting bottlenecks [6], which strongly affect job cycle time and make it very difficult to predict. However, these important features could be detected by mining and analyzing current job cycle time data. Herein, we present a hybrid approach that comprises the fuzzy c-means (FCM)-based genetic algorithm (GA) and a backpropagation network (BPN) to predict job cycle times for semiconductor manufacturing factories.

Semiconductor wafer fabrication is currently a very complex manufacturing process, presenting various production planning and control issues. Predicting the cycle time of each job in a wafer fabrication factory is a critical task for every wafer manufacturer. Glassey

and Resende [7] presented a closed-loop job-release control policy to minimize the average cycle time of jobs in a wafer fabrication factory. Various studies [2,8–10] then emphasized the significance of predicting job cycle times for semiconductor fabrication factories. Here, we identify and review various studies that surveyed job cycle time prediction issues in semiconductor manufacturing.

Initially, Chung and Huang [11] classified the existing approaches used to predict job cycle times in semiconductor manufacturing into four categories, namely (i) statistical, (ii) analytical, (iii) simulation, and (iv) hybrid methods. Now, because of the development of machine learning technology, six categories [2] are generally used, namely (i) statistical, (ii) production simulation (PS), (iii) artificial neural networks (ANNs), (iv) case-based reasoning (CBR), (v) fuzzy modeling, and (vi) hybrid approaches. Herein, we review previous studies in the context of (i) statistical analysis, (ii) analytical methods, (iii) ANNs, (iv) CBR, (v) PS, (vi) fuzzy modeling methods, and (vii) hybrid approaches.

Statistical analysis is a prevalent method in practical applications. A regression model [12] was used to forecast the fab throughput time, including the wait time and processing time. Backus et al. [13] used statistical methods based on modern data-mining algorithms to develop nonlinear predictors for estimating the cycle time of a target lot in a factory. Yang et al. [14] proposed a nonlinear regression metamodel based on queuing theory to generate cycle-time–throughput curves, where simulation experiments were built up sequentially using a multistage procedure. Subsequently, Yang et al. [15] used a factory simulation to fit the metamodels to determine the parameters of the generalized gamma distribution. Pearn et al. [16] presented a due-date assignment model by modeling the waiting time for each product type using a gamma distribution for the due-date assignment problem in a semiconductor fabrication factory. Tai et al. [17] developed an accurate cycle-time estimation method to satisfy a targeted on-time delivery rate. A statistical approach was used to calculate the cycle time for the sum of multiple Weibull-distributed waiting times in the multilayer semiconductor final testing process.

Analytical methods are attracting considerable attention as a way to estimate job cycle times for semiconductor fabrication factories. Chung and Huang [11] provided an analytical approach to developing cycle-time estimation algorithms for engineering lots by analyzing the material flow characteristics in a wafer fabrication factory. Shanthikumar et al. [18] developed a novel solution to reduce the cycle time for each lot in a semiconductor fabrication factory by relaxing a fundamental assumption in classical queuing theory. Morrison and Martin [19] developed a G/G/m-queue model to estimate the total time which a product lot spent in the G/G/m-queue system.

The neural network is one of the most well-known techniques in artificial intelligence [20]. Many studies have also shown that ANN-based methods outperform traditional methods where lengthy and rigorous experiments are avoided while predicting. Chang and Hsieh [21] presented a neural network model to forecast the due date of each order at a wafer fabrication factory, where the experimental results indicated that the proposed approach was effective and efficient when compared with some traditional methods. Sha and Hsu [22] developed an ANN-based due-date assignment rule combined with simulation technology and statistical analysis for predicting lead times at a wafer fabrication factory. Chien et al. [23] developed a manufacturing intelligence approach by integrating Gauss–Newton regression and BPN as a basic model for forecasting the cycle time of a production line. Gazder and Ratrout [24] presented a logit-ANN for mode selection model and applied it in a case study for border transportation. Singh et al. [20] proposed a novel approach for multicriteria decision-making problems using the analytical hierarchy process for evaluating the total transportation cost and an ANN model to investigate the prediction of the total transportation cost. CBR methods have been widely used in previous studies. Chang et al. [1] explored an application of CBR and developed a CBR system using a similarity measure among orders for due-date assignment problems at a wafer fabrication factory. Chiu et al. [25] developed a CBR approach that used a k-nearest-neighbor method with dynamic feature weights and nonlinear similarity functions to predict order due-date

for the due-date assignment problem at a wafer fabrication factory. Chang et al. [26] developed a CBR model in which a GA was used to predict the job cycle time, and a self-organizing map was used to cluster the job cycle time and related shop-floor status at a semiconductor fabrication factory. Liu et al. [27] proposed an approach to predicting job cycle times—by applying evolving fuzzy CBR and self-organizing map methods for a semiconductor fabrication factory.

In addition to the aforementioned approaches, various PS methods have also been proposed for estimating job cycle times for semiconductor fabrication factories. Vig and Dooley [28] presented two new dynamic due-date assignment rules used to predict the job cycle time based on recently completed jobs. Those two new rules were also compared with other established job cycle time–estimation models via computer simulation based on the criterion of due-date performance. Veeger et al. [29] proposed an aggregate model that an extensive simulation study demonstrated could accurately predict the cycle-time distribution of integrated processing workstations in a semiconductor fabrication factory. Hsieh et al. [30] proposed a progressive simulation metamodeling methodology that allowed efficient development of the response surface between the cycle time of regular lots and the percentage of hot lots of high priority in a semiconductor fabrication factory. Yang, Hu [31] built a simulation model for due-date assignment by using the orthogonal kernel least-squares algorithm and imitating the production process of a highly dynamic job shop.

Many fuzzy modeling methods have also been developed previously. Chen [2] developed a fuzzy BPN to incorporate production-control expert judgments with expert opinions to enhance the performance of an existing crisp BPN for predicting the output times of wafer lots. Chang et al. [32] presented a fuzzy modeling method that was evolved further with a GA for the due-date assignment problem in semiconductor manufacturing. Chen and Romanowski [6] proposed a fuzzy data-mining approach based on an innovative fuzzy BPN to determine the lower and upper bounds of job cycle times for wafer fabrication factories. Chen [33] developed a fuzzy neural network-based fluctuation-smoothing rule to better estimate the remaining cycle time of a job at a wafer fabrication factory.

Recently, many hybrid approaches [34,35] based on machine learning and classification have been proposed to improve job cycle time prediction accuracy and accuracy by analyzing the data from semiconductor fabrication factories. Chen [36] applied a fuzzy BPN approach by pre-classifying wafer lots with the k-means classifier before predicting the output time of a job. Chen et al. [37] proposed two hybrid approaches with job record post-classification, namely (i) the equally divided method and (ii) the proportional-to-error method, where a job was post-classified by a BPN after the forecasting error was generated. Tirkel [38] developed cycle-time prediction models by applying machine learning and data-mining methods, where the best neural network model obtained a higher prediction accuracy than that of the decision-tree method. Chen and Lin [39] proposed a fuzzy BPN approach to improve the accuracy of wafer-lot output-time prediction using a relatively small training set. Wang and Zhang [10] designed big-data analytics to predict wafer-lot cycle times. Chen [40] proposed a BPN-based hybrid approach to estimate job cycle times and determine the cycle-time range for a semiconductor manufacturing factory. Chen and Wang [41] estimated the cycle time of a job using a BPN after a nonlinear approach had been used to normalize job cycle times.

Previous studies have used various hybrid approaches for job cycle time prediction [42,43]. Generally, comparisons and trade-offs are made to select a suitable approach among the aforementioned categories. Analytical methods are inappropriate for job cycle time prediction because it is challenging to construct a complex semiconductor manufacturing system analytically. Excessive simulation time and the need for a large amount of data are the main disadvantages of PS [44]. Besides, wafer-lot priorities and machine-dispatching rules change in real-time in semiconductor manufacturing systems, limiting the applicability of statistical methods in predicting the cycle times of wafer lots [10]. ANNs, CBR, and fuzzy modeling methods can provide reasonable prediction accuracy with a rela-

tively small amount of data [44]. Many recent results have shown that classification-based hybrid approaches can improve the accuracy of job cycle time prediction without requiring a large amount of data from semiconductor fabrication factories [44,45]. To improve the performance of job cycle time prediction, we develop a new hybrid approach that combines data mining (FCM-based GA) and machine learning (BPN) methods.

The classification method selection is critical because the quality of the resulting classification can indeed affect the accuracy of job cycle time prediction for semiconductor fabrication factories. However, it is not appropriate to classify the job records exclusively into several clusters. In other words, one job record can belong to multiple clusters with different membership values. In that case, different membership values to different clusters indicate the different amount of effects toward the corresponding clusters. Besides, because it is difficult to determine which cluster a job record belongs to when it falls on the border of two adjacent clusters, FCM has been used in prediction models to classify the data into several clusters in advance, an approach that has been used widely in various fields [46–48]. The FCM algorithm uses the sum of the membership value for all clusters, making it sensitive to noise and isolated data. In addition, FCM is essentially a kind of local hill-climbing algorithm, which makes it sensitive to the initial cluster center and easy to converge to a local extremum [49,50]. To overcome these defects, an FCM-based GA has been introduced.

Before applying the FCM-based GA method, all job records are divided into (i) a dataset for clustering and training and (ii) a dataset for testing. We then use an FCM-based GA classifier to classify the first dataset of job records into several clusters because (i) FCM allows for flexibility in classifying job records and reduces the sensitivity to noise and isolated data, and (ii) a GA can improve the FCM performance by preventing cluster centers from converging to local extrema. The cluster center results obtained from the FCM-based GA classifier are fed into the BPN predictor for training. We discovered that the overfitting of the BPN predictor took place when trained by the raw data containing noises. The BPN predictor predicts job cycle times and compares them with the second dataset.

The remainder of this paper is organized as follows. In Section 2, we describe the proposed FCM-based GA and BPN methods for predicting job cycle times for semiconductor manufacturing. Then, Section 3 uses an actual dataset obtained from a semiconductor fabrication factory to test the proposed hybrid approach. Finally, in Section 4, we conclude by showing this study's contributions and future directions.

2. Methodology

In this section, we present our hybrid approach and explain how it predicts the job cycle time. In semiconductor fabrication factories, six attributes are usually considered for determining the final job cycle time, namely: (i) the job size (pieces); (ii) the factory work-in-process (jobs); (iii) the queue length before the bottleneck (jobs); (iv) the queue length on the rout (jobs); (v) the average waiting time of recently completed jobs (hours); and (vi) factory utilization rate [40,51]. These attributes make different contributions to the final cycle time (hours) of a job. Before we explain the approach in detail, we introduce the notation used herein.

2.1. Notation

The notation used in this model is as follows:

Sets	
\mathcal{R}	Dataset of job records for clustering and training, indexed by $i \in \mathcal{R}$
\mathcal{N}	Dataset of job records for testing, indexed by $n \in \mathcal{N}$
\mathcal{B}	Set of job attributes, indexed by $b \in \mathcal{B}$

Parameters	
v_{ib}	Value of attribute b of job record i
ct_i	Cycle time of job record i
w_b	Weighted value of attribute b
G	Number of clusters, indexed by $c, l \in G$
K	Number of attributes.
m	Fuzziness exponent value
Decisions variables	
z_{cb}	c -th cluster center of attribute b
z_c	c -th K -dimensional cluster center
u_{ic}	Membership value of job record i to c -th cluster
et_c	Expected job cycle time of cluster c
Y_n	Cycle time prediction of new job record n

2.2. Fuzzy c-Means Clustering

The FCM clustering is a soft partitioning algorithm, where jobs with similar attribute values are classified into the same cluster. FCM was developed by Dunn [52] and later improved by Bezdek and Dunn [53]. It is used to assign patterns or data to different clusters, where each data point is allowed to belong to several clusters with different membership values. These membership values represent the extent to which each point belongs to each cluster, and they are also used to update the cluster centers. With this FCM clustering method, the number of clusters is pre-determined, and each data point is then assigned to one or more clusters. The FCM algorithm can be seen as a fuzzified version of the k-means clustering algorithm and is based on minimizing an objective function called the c-means function [53,54]. This takes three input parameters, namely: (i) the number of clusters G ; (ii) the fuzziness exponent value $m > 1$; and (iii) the termination tolerance $\phi > 0$. Given a set of job records \mathcal{R} , including six attributes \mathcal{B} , FCM attempts to minimize the objective function J using the following steps.

Since the proposed hybrid FCM-based GA needs an efficient chromosome design, it is crucial to understand the FCM spatially. To help readers understand, an example of FCM is given in Figure 1 with 6 data points in circles, 2 attributes in axes, and 3 cluster centers in rectangles on two-dimensional spaces. The distance d_{ic} is calculated once cluster centers are obtained.

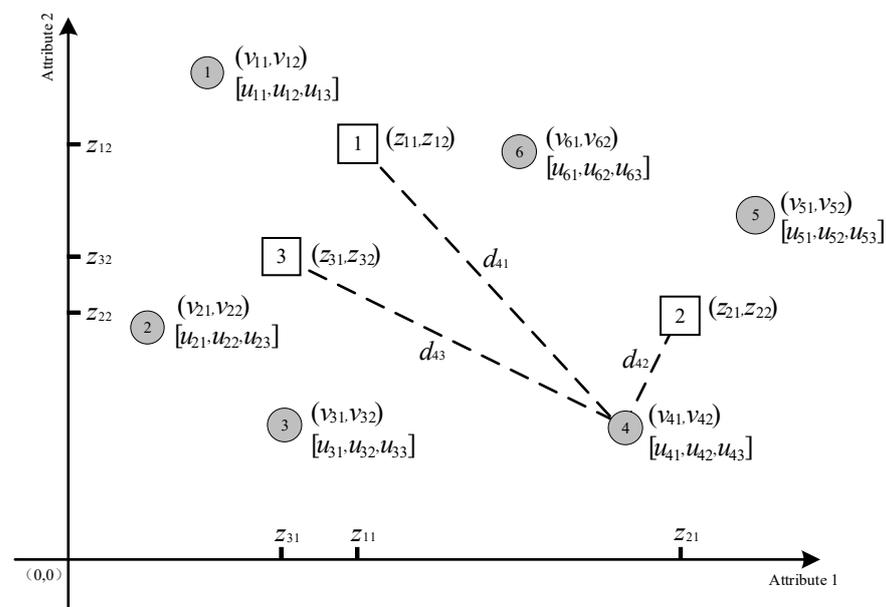


Figure 1. Example of FCM with notation ($G = 3$).

(1) *Minimization of c-means functional* The FCM minimizes an objective function called c-means functional, given as follows. The objective function J measures the sum of fuzzy similarity for job

records \mathcal{R} over all clusters. Note that, at each iteration, the FCM updates the centers of all clusters. The objective function J is given by

$$J = \sum_{c \in G} \sum_{i \in \mathcal{R}} u_{ic}^m \cdot d_{ic}, \tag{1}$$

where d_{ic} is the Euclidean distance between job record i and the center of cluster c .

(2) *Classification (membership updates)* We update the membership value u_{ic} of each job record i toward cluster c . Note that at the start of the algorithm, these membership values u_{ic} are initialized randomly such that $u_{ic} > 0$ and $\sum_{c \in G} u_{ic} = 1$. The fuzziness coefficient $1 < m < \infty$ represents the required clustering tolerance. The membership value u_{ic} is updated by

$$u_{ic} = \frac{1}{\sum_{l \in G} \left(\frac{d_{il}}{d_{ic}}\right)^{\frac{2}{m-1}}}. \tag{2}$$

(3) *Determination of cluster center* We calculate the K-dimensional center for each cluster using Equation (3) and then update the Euclidean distance from each job record i to the center of cluster c , as well as the membership value u_{ic} , using Equation (4):

$$z_{cb} = \sum_{i \in \mathcal{R}} u_{ic}^m \cdot v_{ib} / \sum_{i \in \mathcal{R}} u_{ic}^m, \tag{3}$$

$$d_{ic} = \sqrt{\sum_{b \in \mathcal{B}} [W_b \cdot (v_{ib} - z_{cb})]^2} \tag{4}$$

It is noted that the deviation of values in each attribute is different, and more significant attributes do not overwhelm smaller ones. It was proposed that the original data should be normalized into the same range for all attributes. In this process of reducing prediction errors, w_b can be determined by applying

$$w_b = \frac{100}{\max(v_{\mathcal{R}b}) - \min(v_{\mathcal{R}b})}. \tag{5}$$

(4) *Termination condition* The incremental improvement for objective function value J determines if the iteration continues. When the FCM termination tolerance ϕ is given, the FCM terminates if

$$|J^{s+1} - J^s| < \phi, \tag{6}$$

where s is the iteration number.

2.3. Design of FCM-Based GA

As explained earlier, a GA has been incorporated into FCM to improve the FCM performance by preventing the cluster centers from converging to local extrema too easily. Past research indicates that the combination of GA and FCM methods has apparent strength. Ding and Fu [50] studied a combination of the kernel-based FCM and GA, leading to improved clustering performance. Wikaisuksakul [55] demonstrated that the FCM-NSGA (non-dominated sorting GA) achieves the best partitioning over the other techniques. Ye and Jin [56] suggested a clustering algorithm based on quantum GA, resulting in a better clustering than the general FCM clustering algorithm. Herein, we develop an FCM-based GA approach that combines a GA with FCM. We use center-based string encoding, nonlinear ranking selection, and an adaptive crossover and mutation strategy [55], discussed below.

(1) Chromosome structure

It is always important to have an effective and efficient chromosome structure to the problem in GA. The chromosomes of our proposed FCM-based GA represent the cluster centers by encoding them as a center-based string. Since the job records in this study have K attributes, the K-dimensional centers of G clusters are called genes and are concatenated into a string, as shown in Figure 2. An individual or chromosome comprises G K-dimensional centers. Therefore, the FCM is implemented whenever K-dimensional centers of G clusters need to be found, given attribute values v_{ib} and their membership values u_{ic} to cluster c .

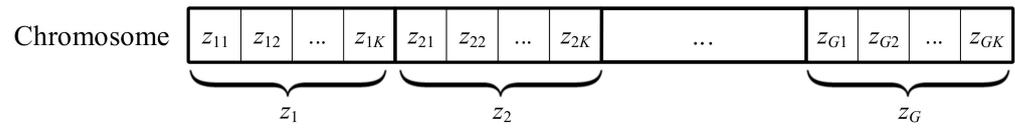


Figure 2. String encoding for cluster centers.

(2) Fitness function

The fitness function is the measure to judge and evaluate individuals (chromosomes) until either a maximum number of generations is reached or the fitness value has converged (i.e., a fitness variance is small enough). In GAs, individuals with higher fitness values are considered better and are more likely to survive. In our case, we used the reciprocal of the objective function as the fitness function to evaluate each individual’s fitness:

$$F = \frac{10^6}{\sum_{c \in G} \sum_{i \in \mathcal{R}} u_{ic}^m \cdot d_{ic}} \tag{7}$$

(3) Genetic operators

The genetic operators that drive the search process are as follows.

(1) Selection operator

Here, the constant-ratio selection method is used to determine individuals to which genetic operations are applied. This selection operator allows us to use good individuals as parents for the population in the next generation. A group is created by selecting a pre-determined percentage (or constant ratio) of individuals from the population, and the best individual among that group is chosen. This process is repeated as many times as the size of the population. This method can preserve the best individual in the current population. With this strategy, fitter individuals have higher survival probabilities, although this does not guarantee that the fittest individual is selected.

(2) Crossover operator

First, we generate a random number α in the interval $[0, 1]$ and compare it with the crossover probability P_c . The crossover operator is applied to two parents that are selected randomly to generate two new children while $\alpha < P_c$. We use a single-point crossover operator, and the crossover point is selected based on a random integer $c \in [1, G]$. The crossover process is shown in Figure 3.

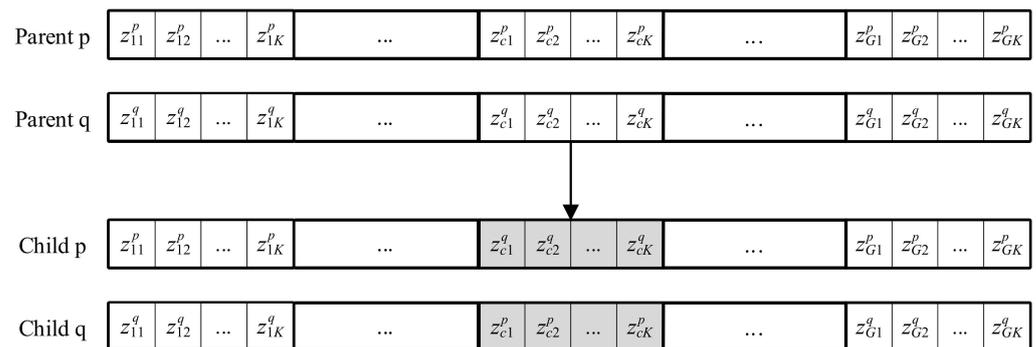


Figure 3. Generating two new children from two parents via crossover.

(3) Mutation operator

For each individual, we generate a random value β in the interval $[0, 1]$ and compare it to the mutation probability P_m . If $\beta < P_m$, we mutate that individual using a single-point mutation one by replacing randomly selected gene with a new random K-dimensional center Z_c^* ($z_{c1}^*, z_{c2}^*, \dots, z_{cK}^*$), as shown in Figure 4.

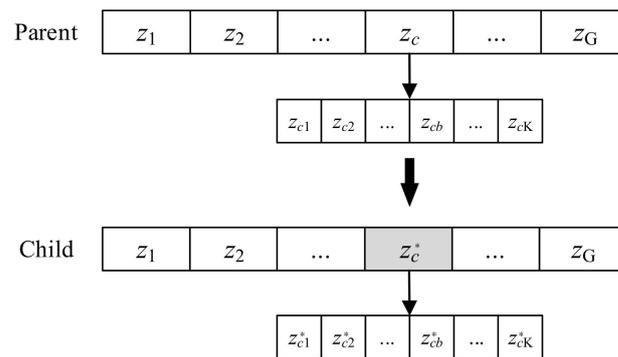


Figure 4. Mutating one individual.

(4) Steps of FCM-based GA

The following steps must be implemented to apply the proposed FCM-based GA.

Step 1: Set the parameter values, namely the number of clusters G , number of generations GN , population size PS , crossover probability P_c , mutation probability P_m , fuzziness value m , and termination tolerance ϵ .

Step 2: Initialize the population. Chromosomes are generated using the FCM clustering methods as many times as PS . To produce a chromosome, we calculate the membership value u_{ic} for each job record i to each cluster c by randomly generated numbers $\gamma_{ic}, 0 \leq \gamma_{ic} \leq 1$, as follows

$$u_{ic} = \gamma_{ic} / \sum_{c \in G} \gamma_{ic}. \tag{8}$$

Step 3: Apply genetic operations. The value of the fitness function F (Equation (7)) can be calculated for each individual. We then use genetic operations, namely the selection, crossover, and mutation operators, to improve population diversity.

Step 4: Apply optimal preservation. For each generation, the fitness values are re-calculated after the genetic operations have been applied to evaluate each individual. Individuals with higher fitness values are more likely to be chosen for survival.

Step 5: Check termination condition. In the present study, the iteration terminates either after a given number of generations GN or when a given fitness variance value ϵ is achieved. If either of these conditions is satisfied, then evolution stops. Otherwise, we return to step 3. For a population of PS , indexed by p , the variance ϵ is calculated as

$$\epsilon = \sum_{p=1}^{PS} \left(F_p - \frac{\sum_{p=1}^{PS} F_p}{PS} \right)^2. \tag{9}$$

After the evolution process is complete, we can obtain the K -dimensional center for each cluster and the membership values. We then associate each job record to all clusters only if its membership value u_{ic} exceeds a given threshold value δ . An auxiliary binary variable ϵ_{ic} is used to calculate the expected job cycle time for all clusters. Finally, the expected job cycle time in each cluster c can be calculated as

$$et_c = \frac{\sum_{i \in \mathcal{R}} u_{ic} \cdot ct_i \cdot \epsilon_{ic}}{\sum_{i \in \mathcal{R}} u_{ic} \cdot \epsilon_{ic}}, \tag{10}$$

$$\text{where, } \epsilon_{ic} = \begin{cases} 1 & \text{if } u_{ic} > \delta \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

Figure 5 shows a flowchart of the complete FCM-based GA approach.

2.4. Backpropagation Network (BPN) Predictor

After the classification is complete, the K -dimensional center for each cluster and the corresponding expected job cycle time is obtained. This information represents the overall behavior and relationship of the dataset. It is important to determine the number of clusters required to capture the most information from the dataset. With too few clusters, it is hard to identify the main features of the dataset. Meanwhile, with too many clusters, the locations of the cluster centers are easily affected by some isolated and noisy job records. Therefore, we test multiple clusters to identify the best combination of experimental parameters and numbers of clusters in the next section.

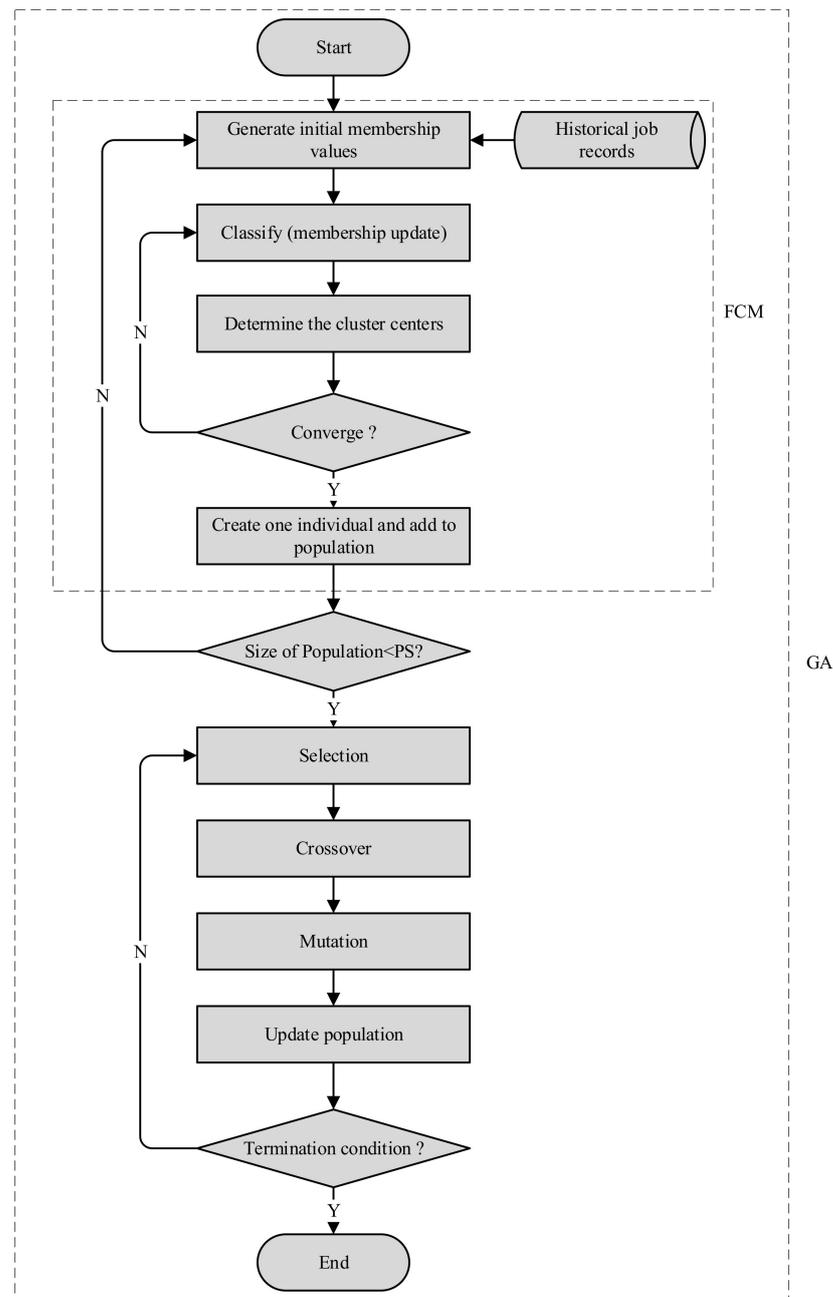


Figure 5. Flowchart of FCM-based GA.

For the cluster centers, the relationship between the independent variables (or attributes) and the corresponding job cycle time has been shown to be nonlinear [23]. As illustrated in Section 2, BPNs are well-known tools for fitting nonlinear relationships. In this sense, a BPN is a good choice for fitting the relationship and predicting the job cycle time [57,58]. Law [59] emphasized that the results predicted by BPNs are accurate with relatively few errors because a BPN adjusts its weights in the output layer to model the training elements. A BPN compares its output with the actual values and propagates the error back through the network. This process is repeated until the error falls within the range of acceptance, whereupon the neural network has been trained successfully. However, determining suitable training and architectural parameters remains difficult. These parameters are usually determined either by trial and error or by pairwise comparisons. The main structure of our BPN is shown in Figure 6.

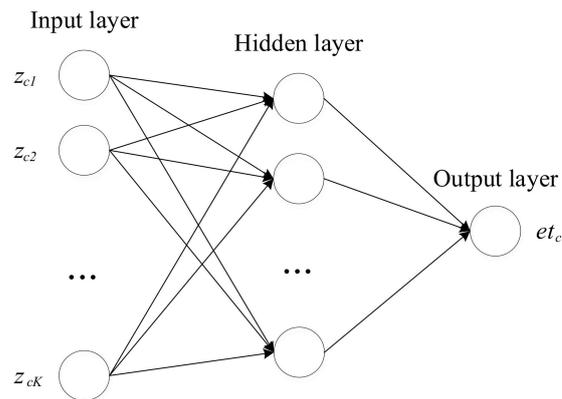


Figure 6. Architecture of the BPN.

Since there is no standard method to construct the BPNs for certain problems [20], we have tested several structures of BPNs on a trial-and-error basis. BPN has a stronger nonlinear mapping ability and flexible network structure. Additionally, the hidden layer may be one or more depending upon the complexity of the problem. More hidden layers might result in overfitting. In this sense, this study uses BPN with one hidden layer rather than ANN for fitting nonlinear relationships between the attribute values and job cycle time. Besides, one hidden layer is used for faster convergence without losing much quality.

The BPN usually consists of at least three layers. Each layer contains many neurons, and the neurons in adjacent layers are interconnected by a set of weights. Here, the detailed configuration and training information of BPN are described as follows:

(1) *Input layer* There are G cluster centers, each with K attributes. Therefore, K neurons are set in the input layer. The G cluster centers with K attribute values and the expected job cycle time are normalized to fall within the interval $[0, 1]$, indexed by τ_{cb} , according to

$$\tau_{cb} = \frac{z_{cb} - \min(v_{\mathcal{R}b})}{\max(v_{\mathcal{R}b}) - \min(v_{\mathcal{R}b})}. \quad (12)$$

(2) *Single hidden layer* The hidden layer may be one or more depends upon the complexity of the problem. In this study, one hidden layer is used for faster convergence and avoiding overfitting without losing much quality.

(3) *Output layer* The predicted job cycle time is obtained in normalized form.

(4) *Training method* Gradient descent has been used as a training method in this study since it is the most commonly used.

(5) *Activation/transformation function* We use the following nonlinear sigmoid function [60] as the activation function:

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (13)$$

where x is a random variable.

(6) *Learning rate* (π): 0.01–1.0.

(7) *Convergence criterion* Many measures can be used to determine when to cease BPN training. Here, the following formula is used in the BPN:

$$T_{\text{error}} = \frac{\sum_{c=1}^G (et_c - \hat{et}_c)^2}{2}, \quad (14)$$

where \hat{et}_c is the job cycle time of cluster c , which is calculated at the output layer using the set of currently trained weights. The BPN training stops when T_{error} falls below 10^{-3} .

3. Computational Results

3.1. Data Description

To demonstrate how the proposed approach is applied, we use a series of 120 job records used in the previous research [51,58]. They were collected from a wafer fabrication factory and are shown in Figure 7. The unit of job cycle time is given by hour (h). The job cycle time has an average of 1237 h and a standard deviation of 205 h. The job cycle time pattern is highly nonstationary and is not stable.

Additionally, the job cycle time is correlated strongly with six attribute variables, as mentioned in the Methodology section. To demonstrate and evaluate the predictive performance of the proposed approach and assess its accuracy fairly, we split these 120 job records into two datasets, namely (i) training dataset and (ii) testing dataset. Since the dataset used for testing usually accounts for a small proportion of the whole dataset, in this study, the first 110 job records are used for (i) the first 110 job records for clustering and training and (ii) the remaining 10 job records for testing.

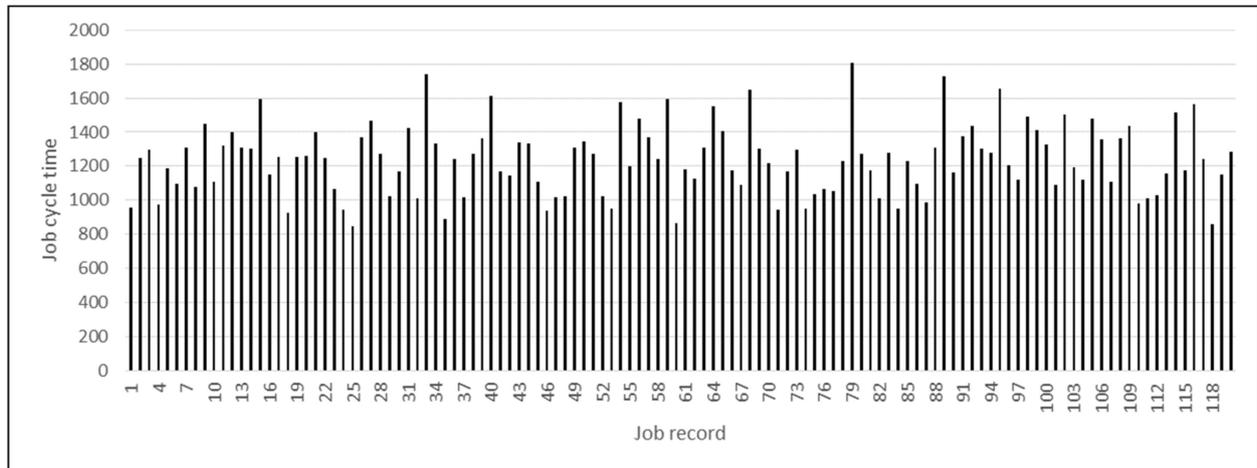


Figure 7. Entire dataset of job records.

3.2. Experimental Settings

The FCM-based GA is used to classify the first 110 job records into G clusters. The obtained cluster centers are fed into the BPN. After learning and training, 10 new testing jobs are given to the trained BPN predictor to predict their cycle times, where the prediction performance is assessed in terms of the mean absolute error (MAE), mean absolute percentage error (MAPE), and root-mean-squared error (RMSE), which are calculated as

$$\text{MAE} = \frac{\sum_{n=1}^{\mathcal{N}} |Y_n - \hat{Y}_n|}{\mathcal{N}}, \quad (15)$$

$$\text{MAPE} = \frac{\sum_{n=1}^{\mathcal{N}} |Y_n - \hat{Y}_n|}{\mathcal{N} \cdot Y_i}, \quad (16)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{\mathcal{N}} (Y_n - \hat{Y}_n)^2}{\mathcal{N}}}, \quad (17)$$

where $|Y_n - \hat{Y}_n|$ is the absolute difference between the predicted and observed values.

Many parameters affect the accuracy of the job cycle time prediction. Herein, we test the performance and compare the results by using different cluster numbers G , different neuron numbers ξ in the hidden layer, and different threshold values δ . In addition, all experiments are conducted using populations with the same number of individuals. The remaining parameters used for the FCM-based GA and BPN are summarized in Table 1.

We implement the proposed hybrid approach in C++ using Visual Studio 2013. The experiments are run on a computer with an Intel Core i7-4790 CPU @3.6 GHz and 16 GB of memory under Windows 10 Professional edition. Because the BPN performance is sensitive to the initial conditions of random weights [39], the experiment is repeated at least five times with different initial conditions. The best and average predicted values are recorded for subsequent analysis.

Table 1. Parameters used in the proposed approach.

Parameter	Value
Fuzziness exponent (m)	2
FCM termination tolerance (ϕ)	10^{-4}
Number of clusters (G)	14, 16, 18, 20, 22, and 24
Population size (PS)	40
Number of generations (GN)	1000
Mutation probability (P_m)	0.2
Crossover probability (P_c)	0.1
Membership threshold value (δ)	0.01, 0.05, 0.08, and 0.1
Fitness variance value (ϵ)	10^{-4}
Learning rate (π)	0.99
Number of neurons in the input layer (φ)	6
Number of neurons in the hidden layer (ξ)	7, 9, 11, 13, and 15

3.3. Experimental Results

The predicted job cycle times and the observed job cycle times for different numbers of clusters and different membership threshold values are presented in Figure 8. It shows that the predicted job cycle time follows similar patterns to those of the observed data. The proposed approach is quite effective in predicting the job cycle time, and computational results show that the observed and predicted job cycle times are matched well when the number of clusters goes from 16 to 20.

Note that the membership threshold value may indeed affect the accuracy of the job cycle time prediction. As the number of clusters increases from 20 to 24, the job cycle time prediction performance fluctuates more under different membership threshold values. Next, we discuss the job cycle time prediction accuracy in terms of the MAE (hours), MAPE (%), and RMSE (hours).

To investigate how the number of clusters affects the prediction accuracy, we experimented using six different numbers of clusters, namely 14, 16, 18, 20, 22, and 24. The MAE, MAPE, and RMSE are used to evaluate the effects of the number of clusters overall testing data for the identical threshold value and the identical number of neurons in the BPN hidden layer, in Figures 9–11, respectively. These figures show similar trends in the MAE, MAPE, and RMSE values against different numbers of clusters, indicating that there might be the need to study optimal numbers of clusters appropriate to datasets but beyond the scope of this study. The curves of MAE, MAPE, and RMSE show that the job cycle time prediction accuracy is strongly correlated with the number of clusters. The job cycle time prediction performance varies with the number of clusters and is poor with too many or too few clusters, meaning that increasing or decreasing the number of clusters does not provide better job cycle time predictions.

In Figure 9, we generally obtain better job cycle time prediction performance (MAE) with the threshold values $\delta = 0.05$ or 0.08 . These threshold values provide good data classification that improves the job cycle time prediction accuracy in the BPN predictor. Similar trends are obtained for the MAPE and RMSE values in Figures 10 and 11, respectively.

To investigate the effect of different numbers of neurons in the BPN hidden layer, we test the job cycle time prediction performance with the identical numbers of clusters (18, 20, and 22) by taking the average MAE, MAPE, and RMSE values for five runs and four threshold values, as shown in Figure 12. The job cycle time prediction performance is clearly not strongly correlated with the number of neurons in the hidden layer of the BPN predictor: increasing the number of neurons ξ improves the prediction accuracy only marginally.

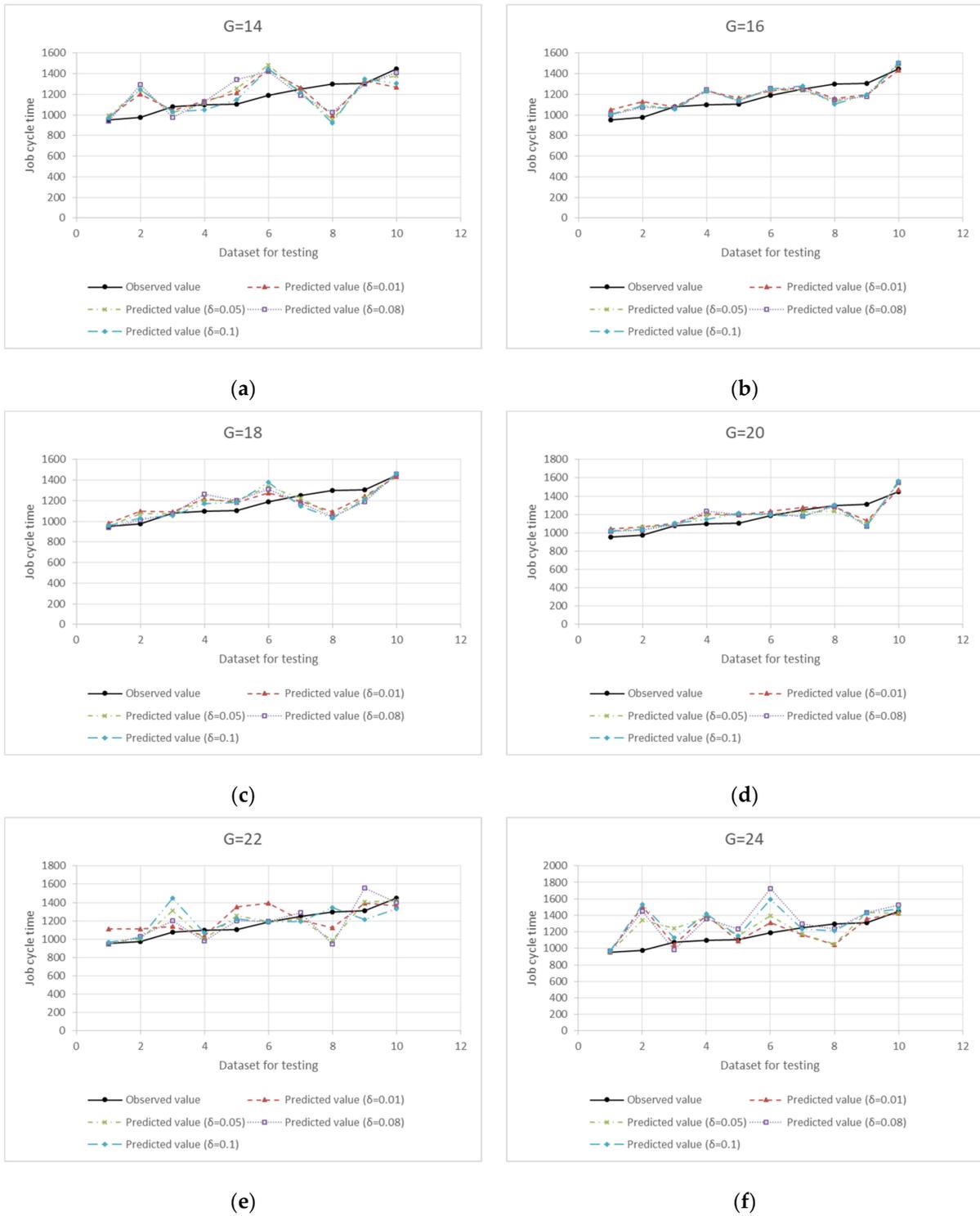


Figure 8. Predicted and observed job cycle times (unit: hour) for different numbers of clusters; 14 (a), 16 (b), 18 (c), 20 (d), 22 (e), and 24 (f).

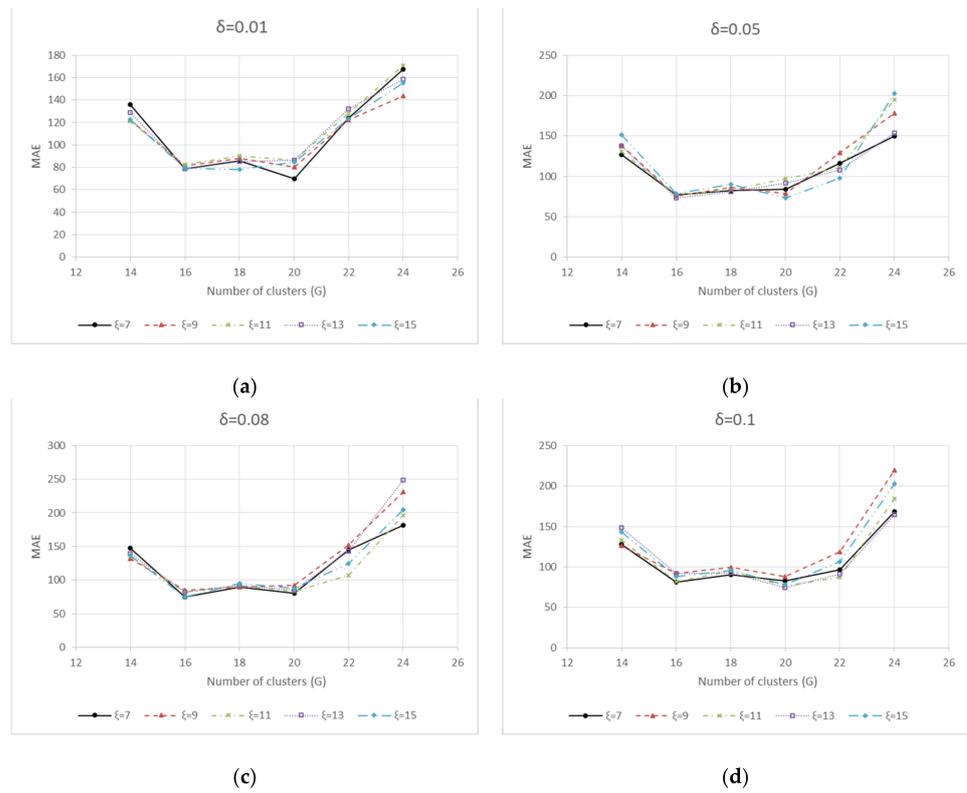


Figure 9. Mean absolute error (MAE, unit: hour) against the number of clusters for different numbers of neurons ξ for different threshold values δ ; $\delta = 0.01$ (a), $\delta = 0.05$ (b) $\delta = 0.08$, (c), and $\delta = 0.1$ (d).

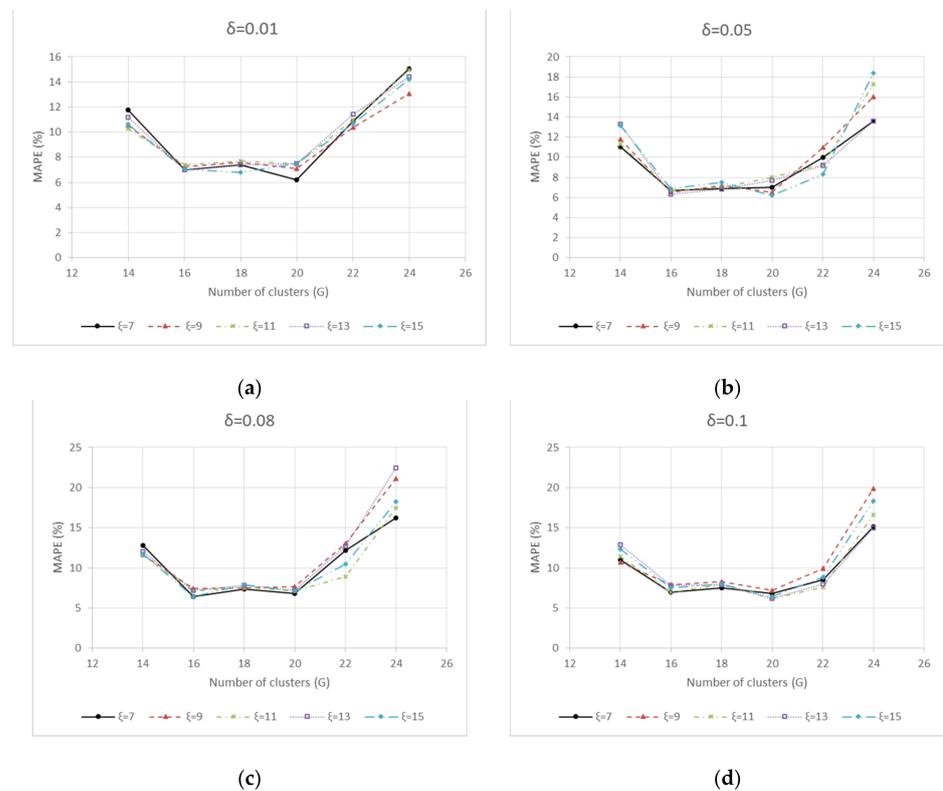


Figure 10. Mean absolute percentage error (MAPE, unit: %) against the number of clusters for different numbers of neurons ξ for different threshold values δ ; $\delta = 0.01$ (a), $\delta = 0.05$ (b) $\delta = 0.08$, (c), and $\delta = 0.1$ (d).

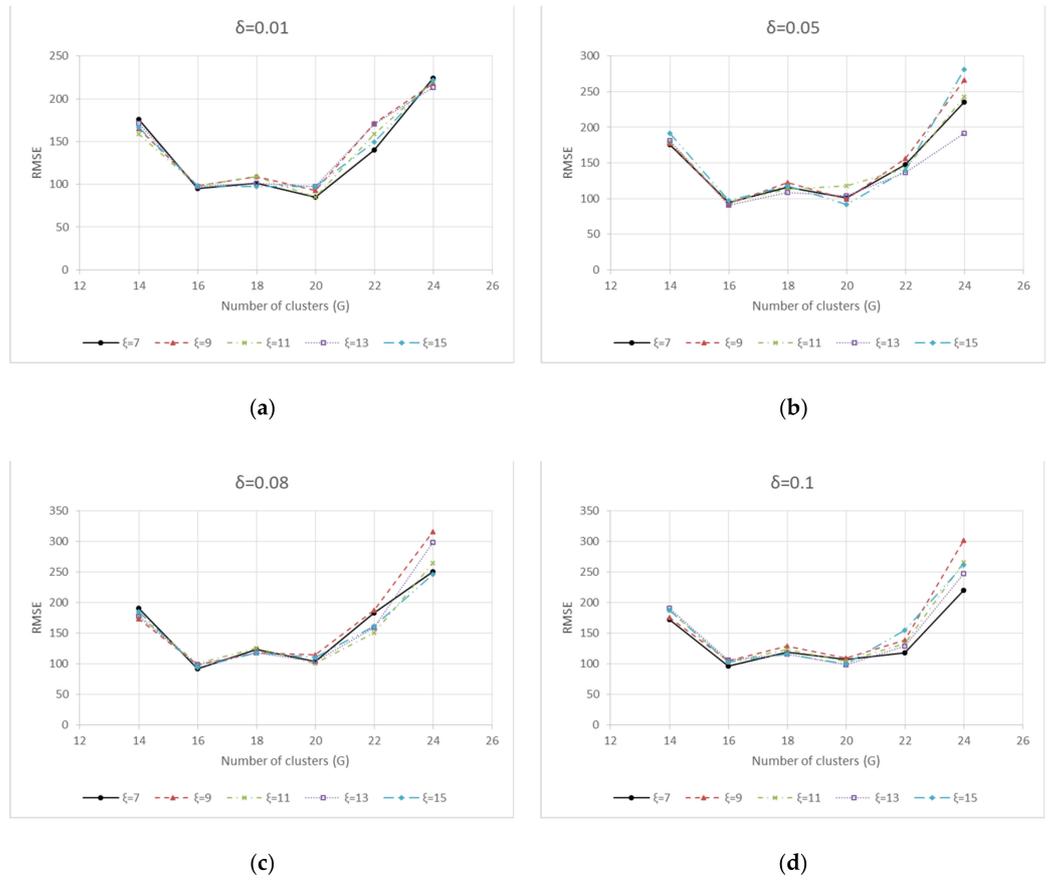


Figure 11. Root-mean-squared error (RMSE, unit: hour) against the number of clusters for different numbers of neurons ξ for different threshold values δ ; $\delta = 0.01$ (a), $\delta = 0.05$ (b) $\delta = 0.08$, (c), and $\delta = 0.1$ (d).

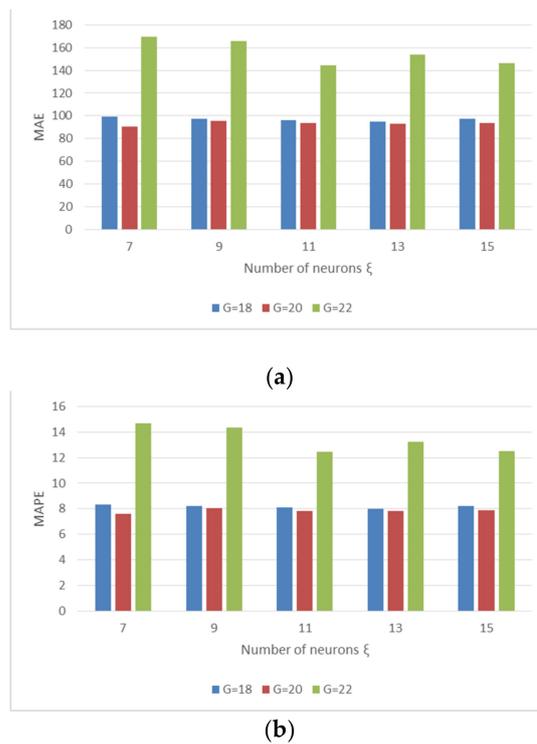
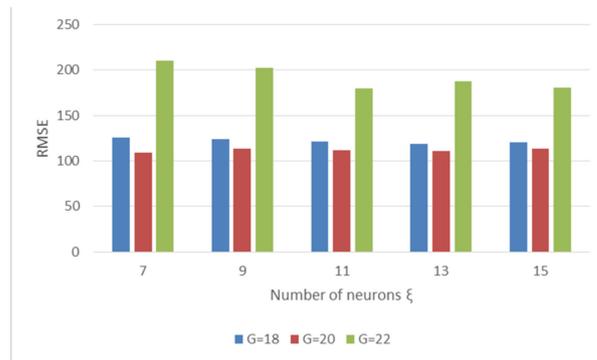


Figure 12. Cont.



(c)

Figure 12. (a) MAAE (unit: hour), (b) MAPE (unit:%), and (c) RMSE (unit: hour) for different numbers of neurons.

Table 2 summarizes the prediction performance, where the best and average MAE, MAPE, and RMSE values are recorded for five experiment runs with three different numbers of clusters (18, 20, and 22), five different numbers of neurons in the hidden layer (7, 9, 11, 13, and 15) and four different threshold value δ (0.01, 0.05, 0.08 and 0.1).

Table 2. Performance of proposed approach in terms of MAE (hours), MAPE, and RMSE (hours).

G.	ξ	δ	Best Value				Average Value			
			0.01	0.05	0.08	0.1	0.01	0.05	0.08	0.1
18	7	MAE (h)	86.04	82.33	89.51	89.99	92.14	105.53	98.87	100.27
		MAPE (%)	7.40	6.90	7.40	7.50	7.90	8.80	8.20	8.30
		RMSE (h)	101.67	115.76	123.37	118.79	111.55	135.14	129.99	125.45
	9	MAE (h)	88.30	86.41	89.53	99.51	93.86	93.99	95.90	106.99
		MAPE (%)	7.60	7.20	7.50	8.30	8.20	7.80	8.00	8.90
		RMSE (h)	108.87	122.07	117.35	128.34	112.11	124.32	127.07	132.91
	11	MAE (h)	90.31	82.96	91.43	95.06	92.97	92.96	97.21	100.36
		MAPE (%)	7.70	7.00	7.60	7.90	8.00	7.80	8.10	8.40
		RMSE (h)	109.41	112.97	125.36	123.62	109.81	119.69	130.09	125.51
	13	MAE (h)	85.39	80.57	92.92	92.87	89.06	92.18	97.22	100.81
		MAPE (%)	7.40	6.80	7.80	7.90	7.70	7.70	8.10	8.40
		RMSE (h)	101.04	108.55	117.95	116.12	105.12	116.75	126.01	127.03
	15	MAE (h)	78.22	89.76	95.48	95.98	92.43	93.11	99.19	103.92
		MAPE (%)	6.80	7.50	7.90	8.00	8.00	7.80	8.30	8.70
		RMSE (h)	97.27	118.15	117.98	115.53	110.32	120.37	125.70	127.18
20	7	MAE (h)	70.03	83.62	80.44	82.74	86.45	90.45	88.62	96.02
		MAPE (%)	6.20	7.00	6.80	6.80	7.60	7.60	7.40	7.90
		RMSE (h)	85.11	101.43	104.14	106.79	98.38	109.19	109.74	118.40
	9	MAE (h)	80.54	78.40	92.96	88.35	88.39	97.85	98.12	98.65
		MAPE (%)	7.10	6.50	7.70	7.20	7.80	8.20	8.10	8.10
		RMSE (h)	93.08	99.15	114.90	108.94	100.88	118.48	117.35	119.08
	11	MAE (h)	86.36	96.98	83.55	76.77	90.25	100.74	97.36	85.25
		MAPE (%)	7.50	8.00	7.10	6.20	7.90	8.30	8.10	7.00
		RMSE (h)	84.78	117.79	100.23	104.06	100.01	120.98	119.32	108.48
	13	MAE (h)	86.17	91.82	85.09	74.43	88.82	94.47	97.58	90.25
		MAPE (%)	7.50	7.70	7.20	6.20	7.70	7.90	8.20	7.50
		RMSE (h)	97.39	104.01	103.59	97.87	100.77	112.05	119.35	113.06
	15	MAE (h)	84.97	73.20	87.94	78.02	91.31	94.05	96.38	93.12
		MAPE (%)	7.50	6.20	7.10	6.40	8.00	7.90	8.00	7.70
		RMSE (h)	97.39	91.56	110.24	99.27	105.69	114.69	118.92	115.74

Table 2. Cont.

G.	ξ	δ	Best Value				Average Value			
			0.01	0.05	0.08	0.1	0.01	0.05	0.08	0.1
7		MAE (h)	123.99	116.03	144.75	96.68	144.20	190.13	187.16	156.08
		MAPE (%)	10.90	10.00	12.20	8.50	12.60	16.70	16.30	13.20
		RMSE (h)	140.28	147.88	182.79	117.55	181.68	238.56	232.60	186.85
9		MAE (h)	122.03	128.79	152.29	118.62	166.13	157.95	179.16	160.69
		MAPE (%)	10.40	11.00	13.00	9.90	14.40	13.70		13.70
		RMSE (h)	171.08	155.78	187.71	138.90	198.50	194.35	222.87	194.79
22	11	MAE (h)	128.49	111.02	107.62	87.14	145.13	141.60	149.09	142.26
		MAPE (%)	11.00	9.30	8.90	7.60	12.40	12.00	13.10	12.20
		RMSE (h)	159.06	142.54	150.50	133.37	177.79	167.46	191.49	180.90
13		MAE (h)	131.87	107.62	143.40	90.92	148.16	156.34	158.13	153.17
		MAPE (%)	11.40	9.20	12.70	8.00	12.70	13.40	13.50	13.30
		RMSE (h)	170.71	136.10	159.68	128.74	185.41	187.92	190.67	185.97
15		MAE (h)	123.17	97.93	124.83	106.08	144.37	133.21	153.99	152.81
		MAPE (%)	10.70	8.30	10.50	8.80	12.40	11.60	13.20	12.90
		RMSE (h)	149.38	139.45	161.30	155.14	175.48	163.15	190.40	194.17

Since the performance measures (MAE, MAPE, and RMSE) are about the prediction errors, the best prediction results were obtained are shaded grey in Table 2 when $G = 20$, $\delta = 0.01$, and $\xi = 7$. Furthermore, to evaluate the effectiveness and efficiency of the proposed hybrid approach, we compare with five existing approaches that were implemented for the same dataset as ours in previous works (Chen 2007b, Chen 2016b); they are (i) linear regression (LR), (ii) BPN, (iii) k -means BPN, (iv) k -means fuzzy BPN [36], and (v) post-classification BPN (Chen [58]).

In Table 3, the best prediction results obtained with the proposed approach with $G = 20$, $\delta = 0.01$, and $\xi = 7$ are compared with those of the five existing approaches. The best results in the previous research were obtained by post-classification BPN (Chen 2016b). The comparison for MAE was made through the following equation. For MAPE and RMSE, similar calculations have been performed to calculate the improvements over the best-known methods.

$$\text{Improvement (\%)} = \frac{\text{MAE}(\text{best known method}) - \text{MAE}(\text{proposed method})}{\text{MAE}(\text{best known method})}, \quad (18)$$

Table 3. Final comparison results.

Method	MAE	MAPE (%)	RMSE
LR	104.5	8.2	120.5
BPN	166.1	12.8	203.7
k -means BPN	99.0	7.7	119.0
k -means fuzzy BPN [36]	91.0	6.9	141.0
Chen [58]	83.6	6.5	105.8
Proposed approach in this study	70.0	6.2	85.1
Improvement (%)	16.3	4.6	19.6

As shown in Table 3, the prediction accuracies are improved by 16.3%, 4.6%, and 19.6% as measured in terms of MAE, MAPE, and RMSE, respectively. Comparing the proposed approach with the previous ones in Table 3 suggests that the former is effective and applicable. It is also clear that the classification-based hybrid approaches (k -means BPN, k -means fuzzy BPN, post-classification BPN, and the proposed approach) perform better than the other approaches (LR and BPN). The proposed hybrid approach is better than other BPNs because it is challenging to train the BPN using raw job records because of the overfitting problem. The proposed approach is better than the k -means fuzzy BPN approach because a better classification is obtained using the FCM-based GA method.

4. Conclusions

This study investigated the relationships between job attributes and job cycle time using a dataset obtained from a semiconductor fabrication factory. To enhance the effectiveness of the job cycle time prediction for semiconductor fabrication factories, a new hybrid approach is proposed, and it comprises FCM-based GA and BPN. Many previous studies used pre-classification or post-classification methods. However, such methods have several drawbacks, such as unequal sizes of different clusters. The FCM-based GA represents a new soft unsupervised classification method wherein the GA is used to improve the classification performance of the FCM. Hence, a good set of cluster centers are obtained while balancing the fuzzy memberships of data points to different centers by applying the FCM-based GA, which captures the main features of the dataset. The results obtained from the FCM-based GA are used as the training data and fed into the BPN predictor. The BPN predictor is then trained to predict the cycle time of new jobs with different attributes.

We also presented a case study to demonstrate the effectiveness of the proposed approach. To obtain good job cycle time prediction performance, various parameters in the experiments were tested to achieve the best performance. The obtained results were compared to those of the previous research to show the dominating performance over the existing methodologies. These efforts lead us to the following conclusions: (1) The FCM-based GA can better pre-classify the first dataset of job records, enabling the BPN to produce accurate job cycle time predictions; (2) Stable and fuzzy job-record classification results allow the BPN to predict the job cycle time more accurately, as shown in Table 3; (3) The proposed hybrid approach indicates that classification-based methods are superior to those without job-record classification.

The limitation of this study is explained here. The computational times were not obtained during the experiments since most studies using the single-layer ANN with the back propagation do not measure it meaningfully. More precisely, the computational times can be divided into (1) FCM clustering and GA, and (2) ANN training. The prediction time is almost instant. The computational times for FCM clustering and GA were varied by various parameters, including the number of centers, threshold values, and many GA parameters (shown in Table 1), which requires significant efforts in the design of experiments to optimize the performance. The training time depends on the size of the data and the training algorithm. Since the primary focus of this study was to propose a novel algorithm, the experimental design to understand the performance is beyond this study.

Another limitation is also related to the optimization of the performance. A simple train/test split was used instead of k-fold cross-validation, which may help the best performance of the ANN training. The primary reason is why the goal of this study was to propose a novel methodology, not to optimize the proposed algorithms. In addition, since FCM clustering is used, we believe that its effects may be aggregated as a grouping in k-fold cross-validation. Furthermore, since we used the clustered c-means by the stochastic algorithm (i.e., GA) instead of the raw data, the benefits of the resampling by k-fold cross-validation may not be justified. However, the study on the benefits of the k-fold cross-validation in our proposed algorithm can be studied in the future.

In future work, we plan to explore the following interesting problems. The optimal number of clusters can be identified by further research of preprocessing the dataset. Given their importance, the proposed approach could be modified to predict the lower and upper job cycle time bounds. In addition, fuzzy concepts could be incorporated into the BPN to enhance prediction accuracy. Another interesting direction for future work would be identifying and eliminating invalid job records because such noise data may affect prediction accuracy. All these questions are considered in future research.

Author Contributions: Conceptualization, G.M.L., X.G.; Formal analysis, G.M.L.; Funding acquisition, G.M.L.; Methodology, X.G.; Project administration G.M.L.; Supervision, G.M.L.; Validation, G.M.L.; Writing—original draft, X.G.; Writing—review & editing, G.M.L. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2018R1A2B3008890).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chang, P.-C.; Hsieh, J.-C.; Liao, T.W. A case-based reasoning approach for due-date assignment in a wafer fabrication factory. In Proceedings of the International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, BC, Canada, 30 July–2 August 2001; pp. 648–659.
2. Chen, T. A fuzzy back propagation network for output time prediction in a wafer fab. *Appl. Soft Comput.* **2003**, *2*, 211–222. [[CrossRef](#)]
3. Kulkarni, V.; Nicola, V.; Smith, R.; Trivedi, K. Numerical Evaluation of Performability and Job Completion Time in Repairable Fault-tolerant Systems. In Proceedings of the 16th Intl. Symp. on Fault Tolerant Computing, Vienna, Austria, 1–4 July 1986.
4. Mehrotra, K.; Chai, J.; Pillutla, S. A study of approximating the moments of the job completion time in PERT networks. *J. Oper. Manag.* **1996**, *14*, 277–289. [[CrossRef](#)]
5. Chen, T. Job cycle time estimation in a wafer fabrication factory with a bi-directional classifying fuzzy-neural approach. *Int. J. Adv. Manuf. Technol.* **2011**, *56*, 1007–1018. [[CrossRef](#)]
6. Chen, T.; Romanowski, R. Precise and accurate job cycle time forecasting in a wafer fabrication factory with a fuzzy data mining approach. *Math. Probl. Eng.* **2013**, *2013*. [[CrossRef](#)]
7. Glassey, C.R.; Resende, M.G. Closed-loop job release control for VLSI circuit manufacturing. *IEEE Trans. Semicond. Manuf.* **1988**, *1*, 36–46. [[CrossRef](#)]
8. Pai, P.-F.; Lee, C.-E.; Su, T.-H. A daily production model for wafer fabrication. *Int. J. Adv. Manuf. Technol.* **2004**, *23*, 58–63. [[CrossRef](#)]
9. Chen, T. A fuzzy-neural approach for estimating the monthly output of a semiconductor manufacturing factory. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 589–598. [[CrossRef](#)]
10. Wang, J.; Zhang, J. Big data analytics for forecasting cycle time in semiconductor wafer fabrication system. *Int. J. Prod. Res.* **2016**, *54*, 7231–7244. [[CrossRef](#)]
11. Chung, S.-H.; Huang, H.-W. Cycle time estimation for wafer fab with engineering lots. *Iie Trans.* **2002**, *34*, 105–118. [[CrossRef](#)]
12. Raddon, A.; Grigsby, B. Throughput time forecasting model. In Proceedings of the 1997 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop ASMC 97 Proceedings, Cambridge, MA, USA, 10–12 September 1997; pp. 430–433.
13. Backus, P.; Janakiram, M.; Mowzoon, S.; Runger, C.; Bhargava, A. Factory cycle-time prediction with a data-mining approach. *IEEE Trans. Semicond. Manuf.* **2006**, *19*, 252–258. [[CrossRef](#)]
14. Yang, F.; Ankenman, B.; Nelson, B.L. Efficient generation of cycle time-throughput curves through simulation and metamodeling. *Nav. Res. Logist. (NRL)* **2007**, *54*, 78–93. [[CrossRef](#)]
15. Yang, F.; Ankenman, B.E.; Nelson, B.L. Estimating cycle time percentile curves for manufacturing systems via simulation. *INFORMS J. Comput.* **2008**, *20*, 628–643. [[CrossRef](#)]
16. Pearn, W.; Chung, S.; Lai, C. Due-date assignment for wafer fabrication under demand variate environment. *IEEE Trans. Semicond. Manuf.* **2007**, *20*, 165–175. [[CrossRef](#)]
17. Tai, Y.; Pearn, W.; Lee, J. Cycle time estimation for semiconductor final testing processes with Weibull-distributed waiting time. *Int. J. Prod. Res.* **2012**, *50*, 581–592. [[CrossRef](#)]
18. Shanthikumar, J.G.; Ding, S.; Zhang, M.T. Queueing theory for semiconductor manufacturing systems: A survey and open problems. *IEEE Trans. Autom. Sci. Eng.* **2007**, *4*, 513–522. [[CrossRef](#)]
19. Morrison, J.R.; Martin, D.P. Practical extensions to cycle time approximations for the $G/G/m$ -queue with applications. *IEEE Trans. Autom. Sci. Eng.* **2007**, *4*, 523–532. [[CrossRef](#)]
20. Singh, A.; Das, A.; Bera, U.K.; Lee, G.M. Prediction of transportation costs using trapezoidal neutrosophic fuzzy analytic hierarchy process and artificial neural networks. *IEEE Access* **2021**, *9*, 103497–103512. [[CrossRef](#)]
21. Chang, P.C.; Hsieh, J.C. A neural networks approach for due-date assignment in a wafer fabrication factory. *Int. J. Ind. Eng.* **2003**, *10*, 55–61.
22. Sha, D.Y.; Hsu, S.Y. Due-date assignment in wafer fabrication using artificial neural networks. *Int. J. Adv. Manuf. Technol.* **2004**, *23*, 768–775. [[CrossRef](#)]
23. Chien, C.-F.; Hsu, C.-Y.; Hsiao, C.-W. Manufacturing intelligence to forecast and reduce semiconductor cycle time. *J. Intell. Manuf.* **2012**, *23*, 2281–2294. [[CrossRef](#)]
24. Gazder, U.; Ratrouf, N.T. A new logit—Artificial neural network ensemble for mode choice modeling: A case study for border transport. *J. Adv. Transp.* **2016**, *49*, 855–866. [[CrossRef](#)]

25. Chiu, C.; Chang, P.-C.; Chiu, N.-H. A case-based expert support system for due-date assignment in a wafer fabrication factory. *J. Intell. Manuf.* **2003**, *14*, 287–296. [[CrossRef](#)]
26. Chang, P.-C.; Fan, C.Y.; Wang, Y.-W. Evolving CBR and data segmentation by SOM for flow time prediction in semiconductor manufacturing factory. *J. Intell. Manuf.* **2009**, *20*, 421. [[CrossRef](#)]
27. Liu, C.-H.; Chang, P.-C.; Kao, I.-W. Cluster based evolving FCBR for flow time prediction in semiconductor manufacturing factory. In Proceedings of the 8th WSEAS International Conference on Applied Computer Science (ACS'08), Venice, Italy, 21–23 November 2008; pp. 424–429.
28. Vig, M.M.; Dooley, K.J. Dynamic rules for due-date assignment. *Int. J. Prod. Res.* **1991**, *29*, 1361–1377. [[CrossRef](#)]
29. Veeger, C.; Etman, L.; Lefeber, E.; Adan, I.; Van Herk, J.; Rooda, J. Predicting cycle time distributions for integrated processing workstations: An aggregate modeling approach. *IEEE Trans. Semicond. Manuf.* **2011**, *24*, 223–236. [[CrossRef](#)]
30. Hsieh, L.Y.; Chang, K.-H.; Chien, C.-F. Efficient development of cycle time response surfaces using progressive simulation metamodeling. *Int. J. Prod. Res.* **2014**, *52*, 3097–3109. [[CrossRef](#)]
31. Yang, D.; Hu, L.; Qian, Y. Due Date Assignment in a Dynamic Job Shop with the Orthogonal Kernel Least Squares Algorithm. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2017; Volume 212.
32. Chang, P.-C.; Hieh, J.-C.; Liao, T.W. Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory. *J. Intell. Manuf.* **2005**, *16*, 549–557. [[CrossRef](#)]
33. Chen, T. Fuzzy-neural-network-based fluctuation smoothing rule for reducing the cycle times of jobs with various priorities in a wafer fabrication plant: A simulation study. *J. Eng. Manuf.* **2009**, *223*, 1033–1043. [[CrossRef](#)]
34. Gao, X.; Zhou, Y.; Amir, M.I.H.; Rosyidah, F.A.; Lee, G.M. A hybrid genetic algorithm for multi-emergency medical service center location-allocation problem in disaster response. *Int. J. Ind. Eng. Theory Appl. Pract.* **2017**, *24*, 663–679.
35. Gao, X.; Lee, G.M. Moment-based Rental Prediction for Bicycle-sharing Transportation Systems Using a Hybrid Genetic Algorithm and Machine Learning. *Comput. Ind. Eng.* **2019**, *128*, 60–69. [[CrossRef](#)]
36. Chen, T. An intelligent hybrid system for wafer lot output time prediction. *Adv. Eng. Inform.* **2007**, *21*, 55–65. [[CrossRef](#)]
37. Chen, T.; Wu, H.-C.; Wang, Y.-C. Fuzzy-neural approaches with example post-classification for estimating job cycle time in a wafer fab. *Appl. Soft Comput.* **2009**, *9*, 1225–1231. [[CrossRef](#)]
38. Tirkel, I. Cycle time prediction in wafer fabrication line by applying data mining methods. In Proceedings of the 2011 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, Saratoga Springs, NY, USA, 16–18 May 2011; pp. 1–5.
39. Chen, T.; Lin, Y.-C. A fuzzy back propagation network ensemble with example classification for lot output time prediction in a wafer fab. *Appl. Soft Comput.* **2009**, *9*, 658–666. [[CrossRef](#)]
40. Chen, T. Asymmetric cycle time bounding in semiconductor manufacturing: An efficient and effective back-propagation-network-based method. *Oper. Res.* **2016**, *16*, 445–468. [[CrossRef](#)]
41. Chen, T.; Wang, Y.-C. A nonlinearly normalized back propagation network and cloud computing approach for determining cycle time allowance during wafer fabrication. *Robot. Comput. Integr. Manuf.* **2017**, *45*, 144–156. [[CrossRef](#)]
42. Yokota, T.; Gen, M.; Li, Y.-X. Genetic algorithm for non-linear mixed integer programming problems and its applications. *Comput. Ind. Eng.* **1996**, *30*, 905–917. [[CrossRef](#)]
43. Yu, X.; Gen, M. *Introduction to Evolutionary Algorithms*; Springer: London, UK, 2010.
44. Chen, T. Incorporating fuzzy c-means and a back-propagation network ensemble to job completion time prediction in a semiconductor fabrication factory. *Fuzzy Sets Syst.* **2007**, *158*, 2153–2168. [[CrossRef](#)]
45. Chen, T. A job-classifying and data-mining approach for estimating job cycle time in a wafer fabrication factory. *Int. J. Adv. Manuf. Technol.* **2012**, *62*, 317–328. [[CrossRef](#)]
46. Chen, T. A hybrid fuzzy-neural approach to job completion time prediction in a semiconductor fabrication factory. *Neurocomputing* **2008**, *71*, 3193–3201. [[CrossRef](#)]
47. Sun, B.; Guo, H.; Karimi, H.R.; Ge, Y.; Xiong, S. Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series. *Neurocomputing* **2015**, *151*, 1528–1536. [[CrossRef](#)]
48. Rezaee, M.J.; Jozmaleki, M.; Valipour, M. Integrating dynamic fuzzy C-means, data envelopment analysis and artificial neural network to online prediction performance of companies in stock exchange. *Phys. A Stat. Mech. Its Appl.* **2018**, *489*, 78–93. [[CrossRef](#)]
49. Biju, V.; Mythili, P. A genetic algorithm based fuzzy C mean clustering model for segmenting microarray images. *Int. J. Comput. Appl.* **2012**, *52*, 42–48.
50. Ding, Y.; Fu, X. Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing* **2016**, *188*, 233–238. [[CrossRef](#)]
51. Chen, T.; Wu, H.-C. A new cloud computing method for establishing asymmetric cycle time intervals in a wafer fabrication factory. *J. Intell. Manuf.* **2017**, *28*, 1095–1107. [[CrossRef](#)]
52. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **1973**, *3*, 32–57. [[CrossRef](#)]
53. Bezdek, J.C.; Dunn, J.C. Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions. *IEEE Trans. Comput.* **1975**, *100*, 835–838. [[CrossRef](#)]

54. Kenesei, T.; Balasko, B.; Abonyi, J. A MATLAB toolbox and its web based variant for fuzzy cluster analysis. In Proceedings of the 7th International Symposium on Hungarian Researchers on Computational Intelligence, Budapest, Hungary, 24–25 November 2006; pp. 24–25.
55. Wikaisuksakul, S. A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering. *Appl. Soft Comput.* **2014**, *24*, 679–691. [[CrossRef](#)]
56. Ye, A.-X.; Jin, Y.-X. A fuzzy c-means clustering algorithm based on improved quantum genetic algorithm. *Simulation* **2016**, *9*, 227–236. [[CrossRef](#)]
57. Chen, T. A systematic cycle time reduction procedure for enhancing the competitiveness and sustainability of a semiconductor manufacturer. *Sustainability* **2013**, *5*, 4637–4652. [[CrossRef](#)]
58. Chen, T. Estimating job cycle time in a wafer fabrication factory: A novel and effective approach based on post-classification. *Appl. Soft Comput.* **2016**, *40*, 558–568. [[CrossRef](#)]
59. Law, R. Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. *Tour. Manag.* **2000**, *21*, 331–340. [[CrossRef](#)]
60. Ripley, B.D. *Pattern Recognition and Neural Networks*; Cambridge University Press: Cambridge, UK, 2007.