

Article

## Enhancing the Effectiveness of Cycle Time Estimation in Wafer Fabrication-Efficient Methodology and Managerial Implications

Toly Chen \* and Yu-Cheng Wang

Department of Industrial Engineering and Systems Management, Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung City 407, Taiwan; E-Mail: tony.cobra@msa.hinet.net

\* Author to whom correspondence should be addressed; E-Mail: tolychen@ms37.hinet.net; Tel.: +886-4-2451-7250.

Received: 19 May 2014; in revised form: 28 July 2014 / Accepted: 28 July 2014 /

Published: 11 August 2014

---

**Abstract:** Cycle time management plays an important role in improving the performance of a wafer fabrication factory. It starts from the estimation of the cycle time of each job in the wafer fabrication factory. Although this topic has been widely investigated, several issues still need to be addressed, such as how to classify jobs suitable for the same estimation mechanism into the same group. In contrast, in most existing methods, jobs are classified according to their attributes. However, the differences between the attributes of two jobs may not be reflected on their cycle times. The bi-objective nature of classification and regression tree (CART) makes it especially suitable for tackling this problem. However, in CART, the cycle times of jobs of a branch are estimated with the same value, which is far from accurate. For these reason, this study proposes a joint use of principal component analysis (PCA), CART, and back propagation network (BPN), in which PCA is applied to construct a series of linear combinations of original variables to form new variables that are as unrelated to each other as possible. According to the new variables, jobs are classified using CART before estimating their cycle times with BPNs. A real case was used to evaluate the effectiveness of the proposed methodology. The experimental results supported the superiority of the proposed methodology over some existing methods. In addition, the managerial implications of the proposed methodology are also discussed with an example.

**Keywords:** cycle time; estimation; classification and regression tree; back propagation network; wafer fabrication

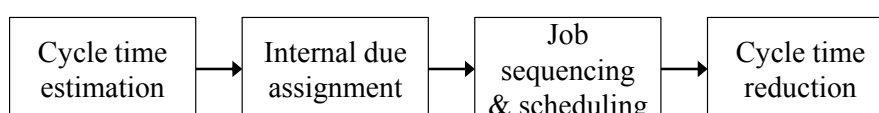
---

## 1. Introduction

Wafer fabrication is a complex and time-consuming process. First, photoresist patterns are photo-masked onto the surface of a wafer. Then, the wafer is exposed to short-wave ultraviolet light. The unexposed areas are etched away and cleaned. Subsequently, hot chemical vapors are deposited onto the desired zones, so that ions can be implanted to form specific patterns at the required depth. These steps are repeated hundreds of times, depending on the complexity of the desired circuits and connections that are usually measured in fractions of micrometers.

This study aims to estimate the cycle time of a job in a wafer fabrication factory. The cycle time (flow time, manufacturing lead time) of a job is the time required for the job to go through the factory. Therefore, it is subject to future demand, capacity constraints, the factory congestion level, the quality of job scheduling, and many other factors. As a result, the cycle time of a job is highly uncertain. According to the competitive semiconductor manufacturing (CSM) survey, the best-performing wafer fabrication factory of memory products achieved an average cycle time of about two days per layer of circuitry [1]. Cycle time management activities include cycle time estimation, internal due date assignment, job sequencing and scheduling, and cycle time reduction (see Figure 1). In practice, shortening the cycle times of jobs is considered an effective way to improve the responsiveness to changes in demand [2]. In addition, according to [3], the number of defects per die has a positive relationship with the cycle time, which means reducing the cycle time can improve product quality. Further, estimating the cycle time of a job helps establish the internal due date for the job. At semiconductor manufacturing factories, the quantities allocated to customers are distributed to daily time buckets according to the available to promise (ATP), which is calculated according to the historical cycle time. For these reasons, estimating and shortening the cycle times of jobs is a very important task to maintain a competitive edge in this industry [4].

**Figure 1.** Cycle time management activities.



In the literature, various types of methods have been proposed to estimate the cycle time of a job in a factory. For example, probability-based statistical methods, such as queuing theory and regression, have been proposed. Furthermore, in these studies, some restrictive assumptions were made, such as exponential processing time distribution [5]. Recently, Pearn *et al.* [6] fitted the waiting time of a job in a wafer fabrication factory with a Gamma distribution. After adding the waiting time to the release time, the cycle time can be derived. This is one of the most important tasks in controlling a wafer fabrication factory. However, the fitted distribution became invalid quickly, making some cycle time estimates far from accurate [7]. Wu [8] constructed a Petri net to estimate the stage cycle time of dual-arm cluster tools with wafer revisiting. Hsieh *et al.* [9] modeled the response surface between the cycle time of normal lots and the percentage of hot lots in semiconductor manufacturing. Chen [10] fitted a fuzzy linear regression (FLR) equation to estimate the cycle time of a job in a wafer fabrication factory. A precise range of the cycle time was also determined. For the same purposes, Chien *et al.* [11]

fitted a nonlinear regression equation instead. Chen [7] applied classification and regression tree (CART) to estimate the cycle time of each job in a wafer fabrication factory. Principal component analysis (PCA) was also applied to generate independent variables from the original ones, which then served as the new inputs to CART.

The application of artificial neural networks (ANNs) is also a mainstream in this field. For example, a self-organization map (SOM) was developed in Chen [12] to classify the jobs in a wafer fabrication factory into a number of categories. Chen [13] and Chang and Hsieh [14] have constructed back propagation networks (BPNs) (or feed-forward neural networks, FNNs) to estimate the cycle time of a job based on the attributes of the job and the current factory conditions. These studies indicated that linear methods are incapable of estimating the cycle time of a job, which supported the application of nonlinear methods such as ANNs. In addition, to improve the effectiveness of an ANN approach, classifying jobs before (or after) estimating the cycle times have been shown to be a viable strategy. To this end, several classifiers were applied, such as k-means (kM) [15], fuzzy c-means (FCM) [16], and SOM [12,17]. A common feature of these classifiers is that all attributes of a job are considered at the same time. In contrast, there are classifiers that consider only some of the job attributes, such as CART and fuzzy inference systems (FISs). The joint use of CART and BPN for estimating the cycle time of a job has rarely been discussed in this field. Chen [18] proposed a BPN tree approach in which the jobs of a branch are separated into two parts for either part a BPN is constructed to estimate the cycle times of jobs. However, Chen's approach relies on extensive and iterative BPN re-learning.

Genetic algorithm (GA) or genetic programming (GP) have also been applied to optimize the parameters of the existing FISs or ANNs to estimate the cycle time of a job, e.g., Chang *et al.* [19], and Nguyen *et al.* [20].

This study proposes a hybrid principal component analysis (PCA), CART, and BPN approach to estimate the cycle time of a job in a wafer fabrication factory. The significance of doing so is six-fold:

- (1) Hybrid algorithms have been shown to be more effective than (pure) algorithms.
- (2) Although variable replacement has been applied to forecasting in many industries, it has not been applied to job cycle time forecasting for semiconductor manufacturers. This study applies PCA to enhance the forecasting performance of the CART-BPN approach.
- (3) In CART, the cycle times of jobs of a branch are estimated with the same value, which is not accurate. In the proposed PCA-CART-BPN approach, a BPN is established for each branch to estimate the cycle times of jobs, which is expected to enhance the estimation accuracy.
- (4) Although clusterwise models, such as SUPPORT and treed Gaussian process models, have been used in various fields, they have not been applied to estimating the cycle time of a job in a manufacturing system.
- (5) Compared with the existing methods, the proposed PCA-CART-BPN approach classify jobs based on fewer job attributes, which is relatively easy to implement. It is also possible to assign more jobs to a branch.
- (6) The existing classifiers in this field, such as kM, FCM, and SOM, classify jobs based on their attributes rather than their compatibilities with the estimation mechanism. However, the compatibility with the estimation mechanism is important, and may be more influential to the estimation performance. In this regard, CART considers the estimation performance in classifying jobs, which makes it more suitable for the same purpose.

In the proposed PCA-CART-BPN approach, first the original variables are replaced according to the results of PCA analysis. Then, jobs are classified using CART. Finally, a BPN is constructed for each category to estimate the cycle times of jobs within the category.

Table 1 is used to compare the proposed PCA-CART-BPN approach with some existing methods in this field. Some methods are easy to use because there have been a lot of software that can seamlessly complete all the necessary steps.

**Table 1.** A comparison of the proposed methodology with some existing methods.

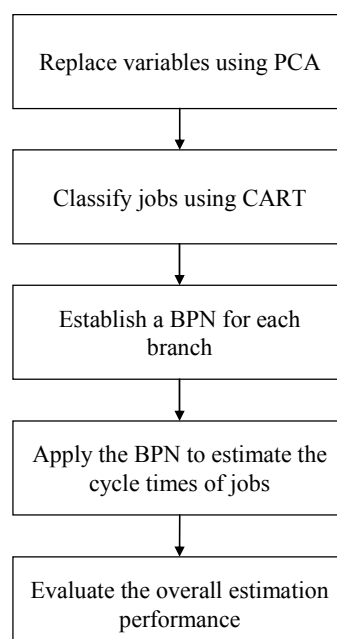
Method	Variable Replacement	Classifier	Estimation Method	Easiness to Use	Accuracy
BPN [14]	-	-	BPN	Easy	Moderate
SOM-BPN [17]	-	SOM	BPN	Moderate	High
Pearn <i>et al.</i> [6]	-	-	Gamma Distribution Fitting	Easy	Low
SOM-FBPN [12]	-	SOM	FBPN	Difficult	High
kM-FBPN [15]	-	kM	FBPN	Difficult	High
PCA-CART [7]	PCA	CART	CART	Easy	Moderate
The proposed methodology	PCA	CART	BPN	Easy	High

The procedure of the proposed PCA-CART-BPN approach is detailed in Section 2, followed by an application to a real case in Section 3. Several existing methods were also applied to the same case for a comparison. Then, the advantages and/or disadvantages of each method were discussed. The managerial implications of the proposed methodology are discussed in Section 4. Finally, Section 5 summarizes the findings of this study, and puts forward some directions that can be explored in future studies.

## 2. Methodology

The proposed methodology is composed of five steps, as illustrated in Figure 2.

**Figure 2.** The procedure for the proposed PCA-CART-BPN approach.



### 2.1. Variable Replacement Using PCA

PCA replaces the original variables by new variables that are independent of each other; these new variables become new inputs to be used in the CART-BPN that estimates job cycle times.

PCA comprises four steps:

- (1) Raw data standardization: The original variables may have excessively large numerical differences and dimensional conflicts. To standardize the dimensions and differences, apply Equations (1) through (3):

$$x_{jk}^* = \frac{x_{jk} - \bar{x}_k}{\sigma_k} \quad (1)$$

$$\bar{x}_k = \frac{\sum_{j=1}^n x_{jk}}{n} \quad (2)$$

$$\sigma_k = \sqrt{\frac{\sum_{j=1}^n (x_{jk} - \bar{x}_k)^2}{n-1}} \quad (3)$$

where  $\bar{x}_i$  and  $\sigma_i$  denote the mean and standard deviation of job attribute  $i$ .

- (2) The correlation matrix  $R$ :

$$R = \frac{1}{n-1} X^{*T} X^* \quad (4)$$

where  $X^*$  is the standardized data matrix. The eigenvalues and eigenvectors of  $R$  are  $\lambda_1 \sim \lambda_m$  and  $u_1 \sim u_m$ , respectively ( $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ ).

- (3) Determination of the number of principal components: The variance contribution rate is:

$$\eta_p = \frac{\lambda_p}{\sum_{q=1}^m \lambda_q} \cdot 100\% \quad (5)$$

and the accumulated variance contribution rate is:

$$\eta_{\Sigma}(p) = \sum_{q=1}^p \eta_q \quad (6)$$

Choose the smallest  $p$  value such that  $\eta_{\Sigma}(p) \geq 85\% \sim 90\%$ .

- (4) Formation of the following matrixes:

$$U_{m \times p} = [u_1, u_2, \dots, u_p] \quad (7)$$

$$Z_{n \times p} = X_{n \times m}^* U_{m \times p} \quad (8)$$

## 2.2. Job Classification Using CART

CART, introduced by Breiman *et al.* [21], is a statistical procedure primarily used for object classification. The objective is to classify a group of objects into two or more populations. CART can handle both categorical and continuous data. CART uses an exhaustive search in splitting with an objective to improve the impurity measure, *i.e.*, the reduction in the residual sum of squares.

The procedure of CART is composed of stages including tree growing, stopping, and pruning. The first stage is to grow the tree using a recursive partitioning technique that selects variables and split points according to a pre-specified criterion. Criteria to this end include Gini, towing, ordered towing, and maximum deviance reduction [21].

Tree growing stops if any of the following conditions is satisfied:

- (1) The improvement in the performance measure has become insignificant with more branches.
- (2) A certain number of nodes have been generated.
- (3) The depth of the tree has reached a certain level.

A large tree may overfit, while a small tree may not reflect the inherent structure of the data. Cost-complexity pruning is usually used to tackle this issue. The cost-complexity of a tree  $T$  is the sum of sum of squared error (SSE) and the penalty on the complexity/size of the tree:

$$C_\alpha(T) = \text{SSE} + \alpha |T| \quad (9)$$

$$\text{SSE} = \sum_j (CT_j - y_j)^2 \quad (10)$$

where  $CT_j$  and  $y_j$  are the cycle time estimate and actual cycle time of job  $j$ , respectively. The results of cost-complexity pruning are a nested subset of trees starting from the largest tree and ending with the smallest tree (with only a single node). The effectiveness of a subtree can be evaluated by cross-validation or using another (testing) data.

In the traditional CART approach, the cycle times of jobs assigned to a branch ( $b$ ) are estimated with the same value  $y_{(b)}$  that is equal to the average of the historical cycle times:

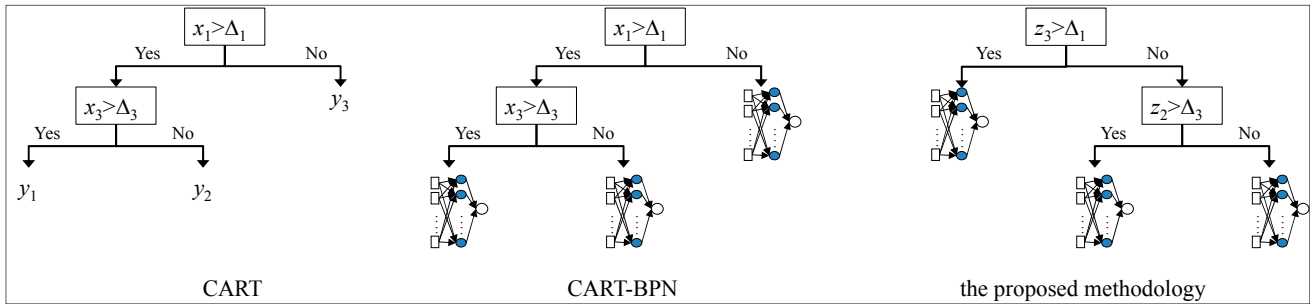
$$y_j = y_{(b)} = \frac{\sum_{o \in G(b)} CT_o}{\sum_{o \in G(b)} 1} \quad \forall j \in G(b) \quad (11)$$

where  $G(b)$  represents the set of jobs of branch  $b$ . However, such a treatment is far from accurate. For this reason, in this study, a BPN is constructed for each branch to estimate the cycle times of jobs assigned to this branch:

$$y_j = \text{BPN}_b(\mathbf{z}_j) \quad (12)$$

where  $\text{BPN}_b$  is the BPN constructed for branch  $b$  to estimate the job cycle times;  $\mathbf{z}_j$  is the vector of the new attributes of job  $j$ . The comparison of CART, CART-BPN, and the proposed methodology is illustrated in Figure 3, in which  $\Delta_k$  is the boundary value used to split the data along the new attribute  $k$ .

Figure 3. Comparison of CART and the proposed methodology.



2.3. Job Cycle Time Estimation within Each Node Using BPN

2.3.1. Rationale

In a wafer fabrication factory, the relationship between the cycle time and attributes of a job has been shown to be a nonlinear one [11]. Consider the simplest example. Sometimes wafers are processed piece by piece, while in other steps tens of wafers are processed as a whole. As a result, the relationship between the cycle time and size of a job cannot be fitted with a linear equation. BPN is a well-known tool for fitting nonlinear relationships, so is CART. A combination of CART and BPN is natural, and has potential for improving the performance of estimating the cycle time of a job.

Another question is why jobs with similar attributes have very different cycle times. That is because most of the attributes of a job were determined when the job was released into the factory. However, the cycle time of a job depends on the future conditions of the factory. That may be much different even for jobs with similar attributes. That explains the incapability of the existing methods based on job classification.

2.3.2. Procedure

Subsequently, a BPN is constructed to estimate the cycle times of jobs. The BPN is configured as follows. There are  $K$  inputs to the BPN including the new attributes of a job. A lot of past studies have shown that a BPN with a single hidden layer can achieve a satisfactory approximation performance [15–17,22]. In addition, several ways have been proposed in the literature to determine the number of neurons in the hidden layer, e.g., [22–24]. In this study, common neural network learning parameters, including number of hidden layers, number of hidden layer units, learning rate, and learning time, all are determined using a trial-and-error method. The activation/transformation functions for the input and hidden layers are the linear activation function and the hyperbolic tangent sigmoid function, respectively:

$$\text{(linear activation)} \quad f(x) = x \tag{13}$$

$$\text{(hyperbolic tangent sigmoid)} \quad f(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

Inputs to the BPN are multiplied by the weights of the connections between the input and hidden layers, then are summed on each neuron in the hidden layer. After being compared with the threshold on the neuron, only significant signals will be transformed and outputted as:

$$h_{jl} = \frac{1}{1 + e^{-n_{jl}^h}} \quad (15)$$

where:

$$n_{jl}^h = I_{jl}^h - \theta_l^h \quad (16)$$

$$I_{jl}^h = \sum_{k=1}^K w_{kl}^h \cdot z_{jk} \quad (17)$$

$h_{jl}$  is the outputted signal from hidden-layer neuron  $l$  for job  $j$ ;  $\theta_l^h$  is the threshold on hidden-layer neuron  $l$ ;  $w_{kl}^h$  is the weight of the connection between input-layer neuron  $k$  and hidden-layer neuron  $l$ .

Signals outputted from the hidden-layer neurons are transmitted to the neuron in the output layer in the same manner. Finally, the output from the BPN is generated as:

$$o_j = \frac{1}{1 + e^{-n_j^o}} \quad (18)$$

where:

$$n_j^o = I_j^o - \theta^o \quad (19)$$

$$I_j^o = \sum_{l=1}^{2K} w_l^o \cdot h_{jl} \quad (20)$$

$\theta^o$  is the threshold on the output-layer neuron;  $w_l^o$  is the weight of the connection between hidden-layer neuron  $l$  and the output-layer neuron.

Subsequently, a lot of algorithms can be applied to train a BPN, such as the gradient descent (GD) algorithm, the conjugate gradient algorithm, the Levenberg-Marquardt (LM) algorithm, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, and others. For a recent comparison of these algorithms, refer to [22]. Among these algorithms, the LM algorithm has a faster convergence speed, therefore is applied in the proposed methodology, as described below.

First, placing the inputs and the BPN parameters in vectors:

$$\mathbf{z}_j = [z_{j1} \dots z_{jK}] \quad (21)$$

and

$$\boldsymbol{\beta} = [w_{11}^h, \dots, w_{KL}^h, \theta_1^h, \dots, \theta_L^h, w_1^o, \dots, w_L^o, \theta^o] \quad (22)$$

respectively. Then, the network output  $o_j$  can be represented with:

$$o_j = f(\mathbf{z}_j, \boldsymbol{\beta}) \quad (23)$$

Substituting Equation (25) into Equation (14),

$$\text{SSE} = \sum_{j=1}^n (N(CT_j) - f(\mathbf{z}_j, \boldsymbol{\beta}))^2 \quad (24)$$

To find out the optimize values of  $\boldsymbol{\beta}$ , an iterative procedure is used in the LM algorithm:



- (1) Specify the initial values of  $\beta$ , e.g.  $\beta = [1, 1, \dots, 1]$ .
- (2) Replace  $\beta$  by  $\beta + \delta$ , where  $\delta = [\Delta w_{11}^h, \dots, \Delta w_{KL}^h, \Delta \theta_1^h, \dots, \Delta \theta_L^h, \Delta w_1^o, \dots, \Delta w_L^o, \Delta \theta^o]$ :

$$o_j = f(z_j, \beta + \delta) \tag{25}$$

- (3) Approximating the right-hand side by its linearization gives:

$$f(z_j, \beta + \delta) \approx f(z_j, \beta) + J_j \delta \tag{26}$$

where:

$$J_j = \partial f(z_j, \beta) / \partial \beta \tag{27}$$

is the gradient vector of  $f$  with respect to  $\beta$ . SSE becomes:

$$SSE \approx \sum_{j=1}^n (N(CT_j) - f(z_j, \beta) - J_j \delta)^2 \tag{28}$$

- (4) The optimal value of  $\delta$  can be obtained by taking the derivative of SSE with respect to  $\delta$  and setting the result to zero. For details refer to [25].
- (5) Return to step (2).

### 3. Applications

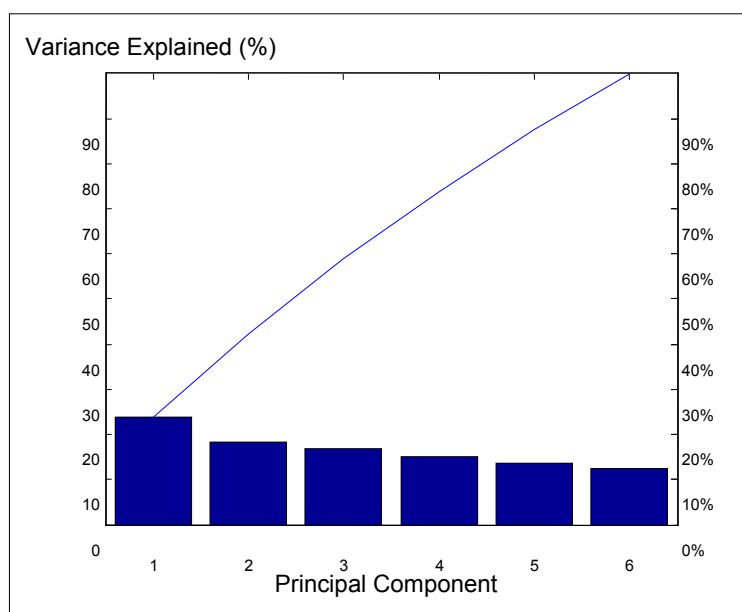
A real case containing the data of 120 jobs from a wafer fabrication factory located in Taichung Scientific Park, Taiwan, was used to evaluate the effectiveness of the proposed methodology (see Table 2). There are tens of dynamic random access memory (DRAM) products in the wafer fabrication factory. Six attributes of a job, including job size, factory utilization, the queue length on the route, the queue length before the bottleneck, work-in-process (WIP), and the average waiting time, are indicated with  $x_j = [x_{j1} \ x_{j2} \ x_{j3} \ x_{j4} \ x_{j5} \ x_{j6}]$ . The average waiting time is the average of the waiting times of the three most recently completed jobs. That measures the extent of delay that each job is likely to face. The six attributes were chosen from about twenty candidates after the backward elimination of regression analysis. The proposed PCA-CART-BPN approach was implemented on a PC with an Intel Dual CPU E2200 2.2 GHz and 2.0G RAM. The BPN was implemented with the Neural Network Toolbox of MATLAB 2006a.

**Table 2.** The collected data.

<b>j</b>	<b>x<sub>j1</sub></b>	<b>x<sub>j2</sub></b>	<b>x<sub>j3</sub></b>	<b>x<sub>j4</sub></b>	<b>x<sub>j5</sub> (h)</b>	<b>x<sub>j6</sub></b>	<b>CT<sub>j</sub> (h)</b>
1	24	1223	158	807	99	84.2%	953
2	23	1225	164	665	142	94.8%	1248
3	25	1232	154	718	373	88.4%	1299
4	23	1282	165	813	148	92.9%	976
5	22	1352	182	760	389	93.1%	1189
6	24	1293	182	760	69	96.2%	1095
				...			
120	22	1319	159	777	326	88.8%	1285

At first, a Pareto comparison of percentage of variability explained by each principal component is shown in Figure 4. An individual component by itself was able to explain less than 30% of the variance; the present study proposed to explain 85%–90% of the variance, and thus  $p$  was set to 5. The five principal components shown in Figure 4 explained approximately 85% of the total variability in the standardized data; therefore, the five-component analysis was a reasonable reduction of dimensions.

**Figure 4.** The Pareto analysis chart.



The coordinates of the original data were calculated in terms of the new coordinate system to produce component scores. Table 3 shows the component scores, which were used as new inputs to the BPN.

**Table 3.** New inputs to the BPN.

$j$	$z_{j1}$	$z_{j2}$	$z_{j3}$	$z_{j4}$	$z_{j5}$
1	0.547	0.855	-1.640	-1.345	-0.591
2	-0.649	0.361	-0.850	0.047	-1.115
3	-0.596	-1.170	-1.984	0.388	-1.015
4	0.218	1.062	-1.090	-0.468	0.885
5	0.296	-0.073	1.178	1.198	1.764
6	-1.318	2.127	0.158	0.214	0.305
			...		
120	0.930	-0.846	-0.677	-0.361	1.238

Subsequently, the new data were divided into two parts—the training data (the first 90 jobs) and the testing data (the remaining 30 jobs) (see Table 2). The training data were not normalized before creating the CART tree, but were normalized into  $[0.1, 0.9]$  before they were learned by the BPNs. In fact, data normalization is not helpful for CART, but is conducive to the fast convergence of the BPN. Only the performance to the testing data was evaluated. In addition, four-fold cross validation was applied to effectively reduce bias and variability.

Subsequently, CART was applied to classify jobs according to the new attributes. The impurity measure in CART was Gini. The popular 1-SE rule was applied for the CART tree pruning. In this way, the smallest tree which cross-validation cost was less than the minimum cross-validation cost plus the standard deviation of the cross-validation cost was chosen. The results are shown in Figure 5.

**Figure 5.** The classification results using PCA-CART. (A) 1st run; (B) 2nd run; (C) 3rd run; (D) 4th run.

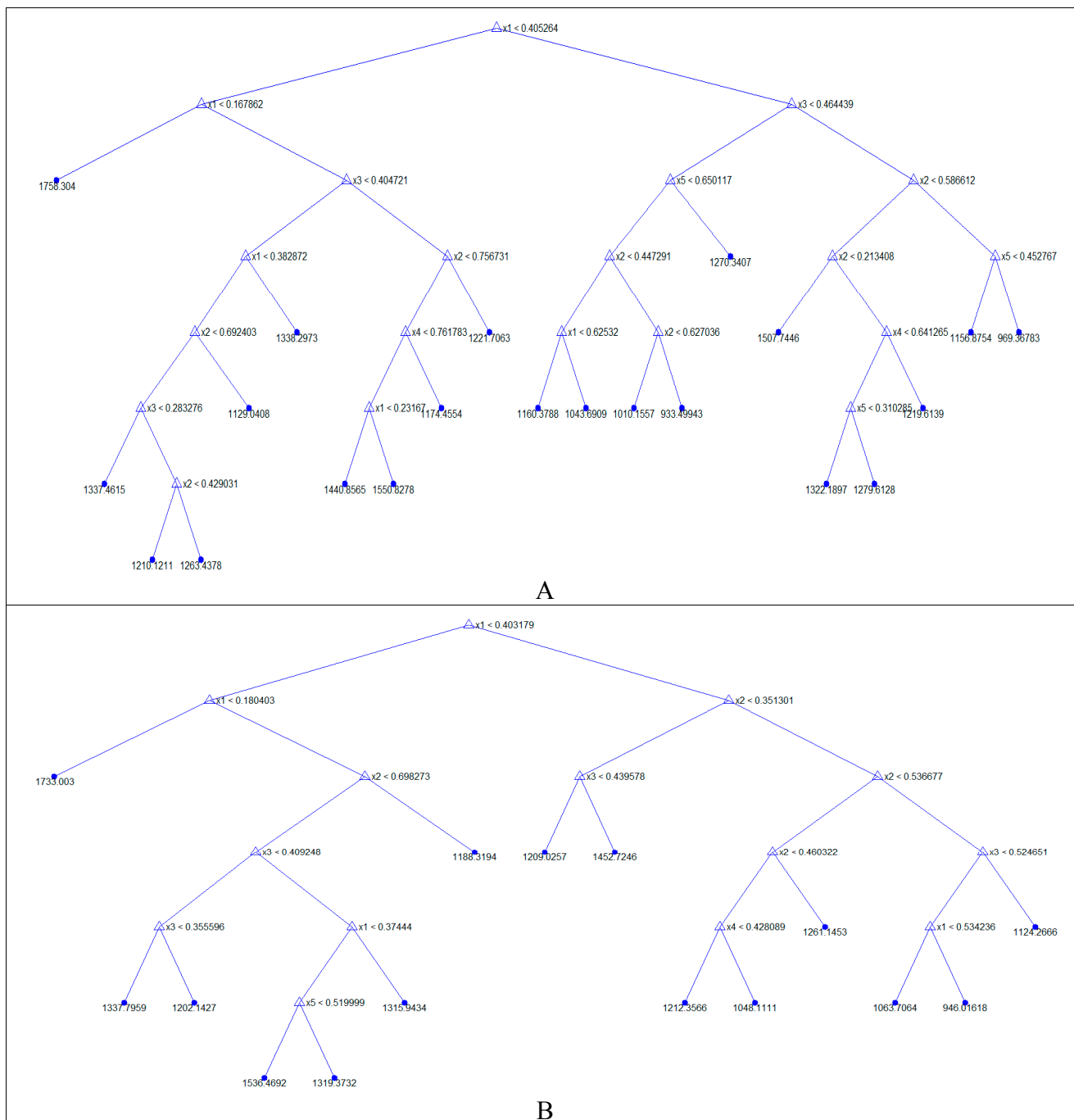
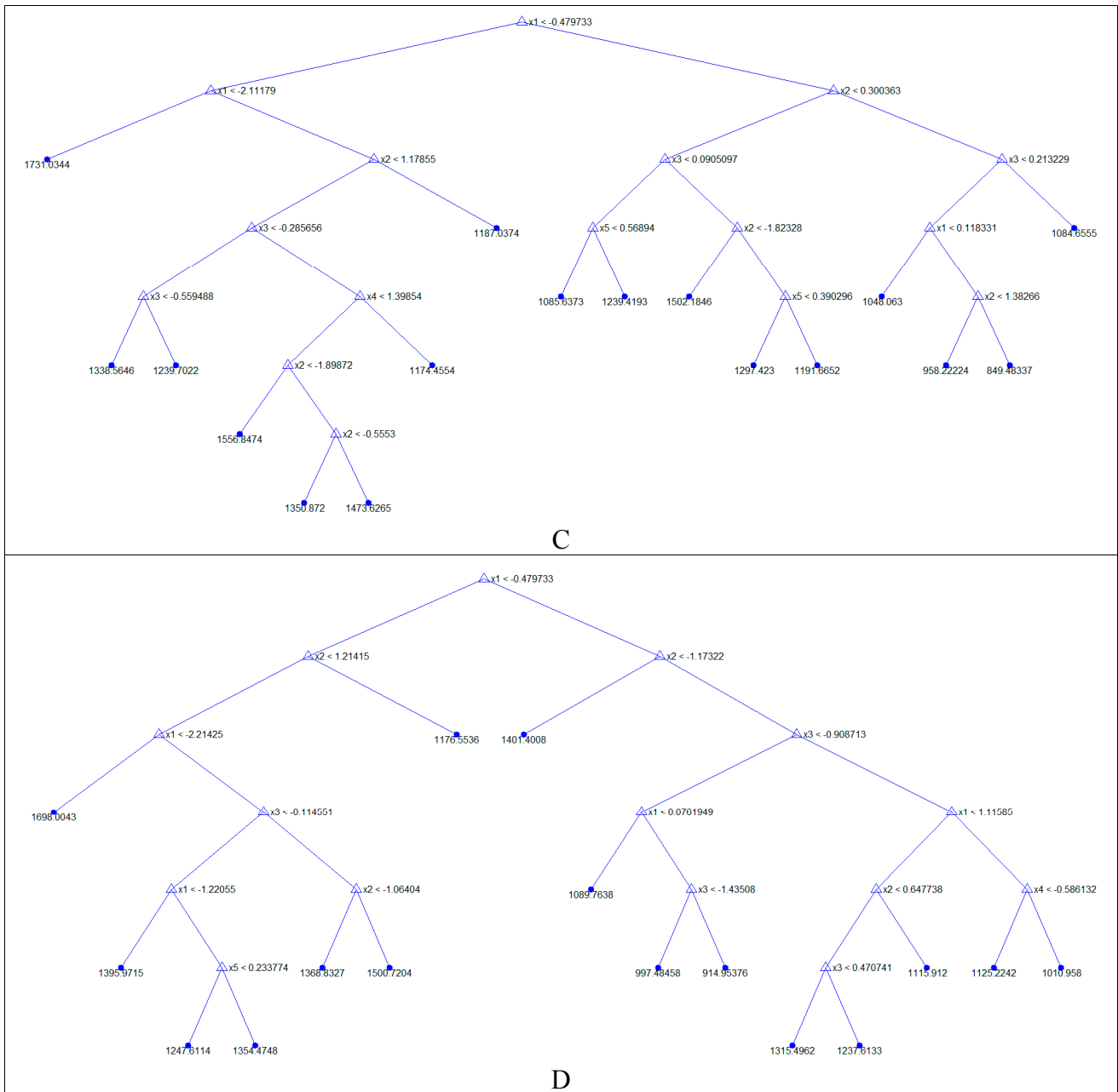


Figure 5. Cont.



Obviously, the number of job groups is much more than that obtained using the existing classifier like kM, FCM, or SOM. The classification results were also different from those using CART alone without PCA (see Figure 6). However, some groups contained too few jobs to train a BPN. For this reason, only for groups with more than five jobs, BPNs were applied to improve the cycle times.

Jobs in each group were used to train the BPN of the group. In this study, a trail-and-error method was used to test the optimal number of neurons. When the learning rate ( $\eta$ ) was 0.1, testing various numbers of neurons affected the prediction error. After repeatedly testing five times, when the number of hidden layer neurons was set to 8, the minimal prediction error (minimal RMSE) was obtained (Figure 7). To define the optimal parameter values of the epochs and learning rate ( $\eta$ ) and to obtain the best prediction results, the changes in the prediction error when using various combinations of learning rates ( $\eta$ ) and epochs were analyzed. The results indicated the minimal RMSE was achieved when the

epochs were 50,000 and the learning rate ( $\eta$ ) was 0.9 (Figure 8). However, that resulted in a very lengthy learning process. In addition, when the learning rate ( $\eta$ ) was 0.9, setting the epochs to 30,000 was a favorable choice. For this reason, in the experiment, the learning rate ( $\eta$ ) and epochs were set to 0.9 and 30,000, respectively.

**Figure 6.** The classification results using CART alone without PCA. (A) 1st run; (B) 2nd run; (C) 3rd run; (D) 4th run.

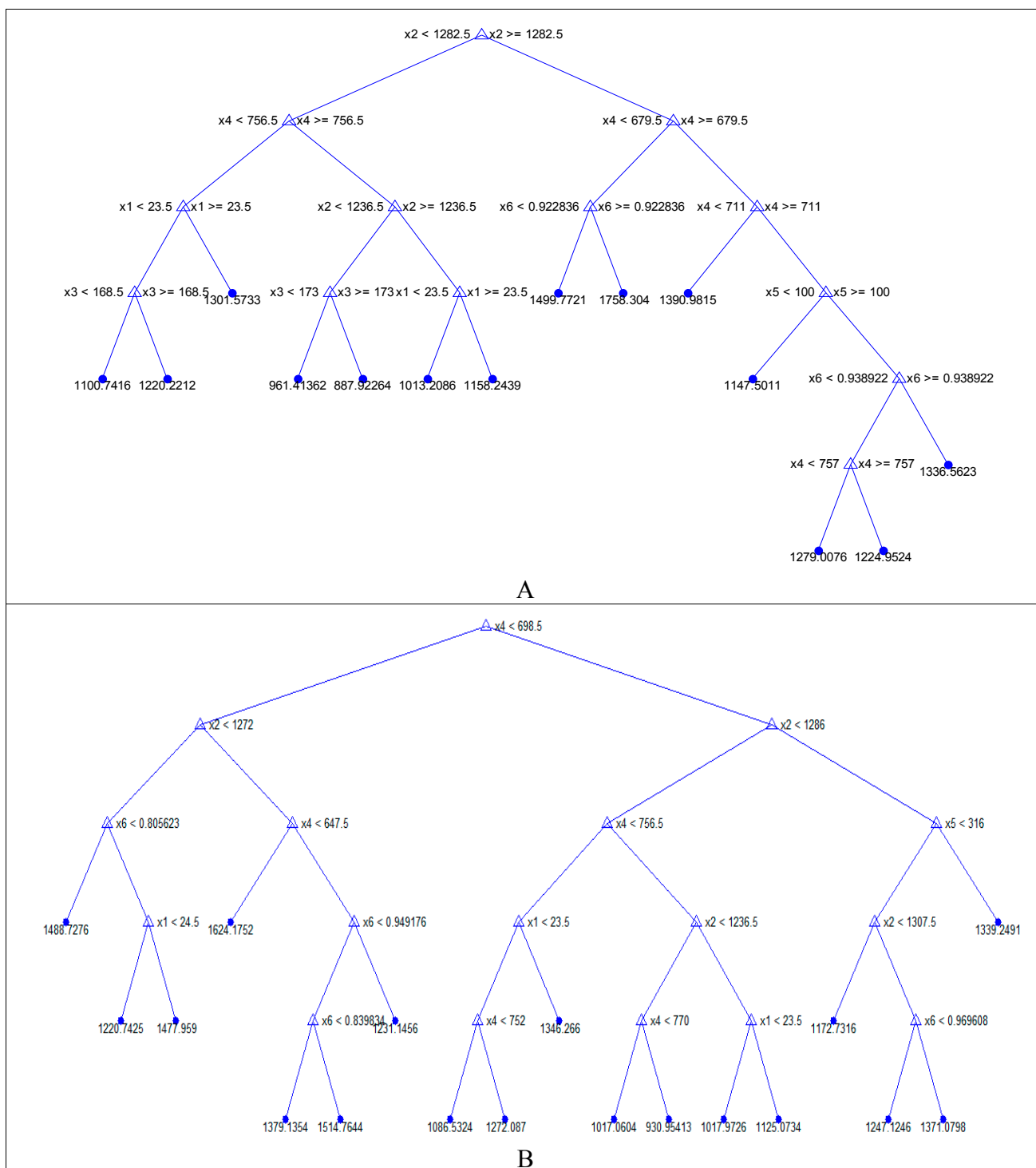
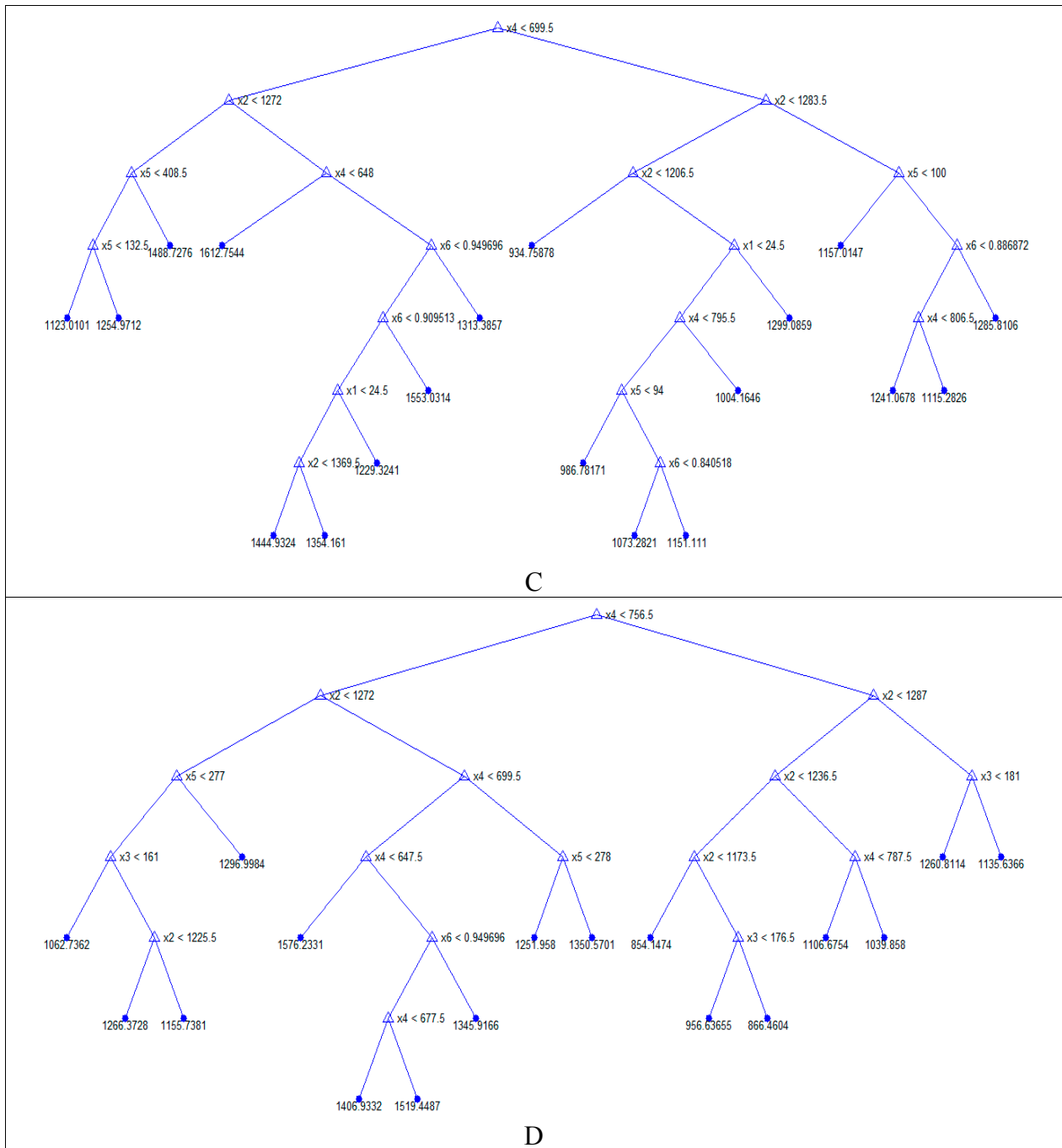


Figure 6. Cont.



After training, the BPN can be used to estimate the cycle times of jobs in the testing data that belong to this group. The estimation accuracy of the PCA-CART-BPN method was evaluated in terms of three criteria: the MAE, mean absolute percentage error (MAPE), and root mean squared error (RMSE). Table 4 summarizes the results. The performances of the CART, CART-BPN, and PCA-BPN methods are shown in Tables 5–7 for a comparison.

Figure 7. Various numbers of neurons affected the prediction error.

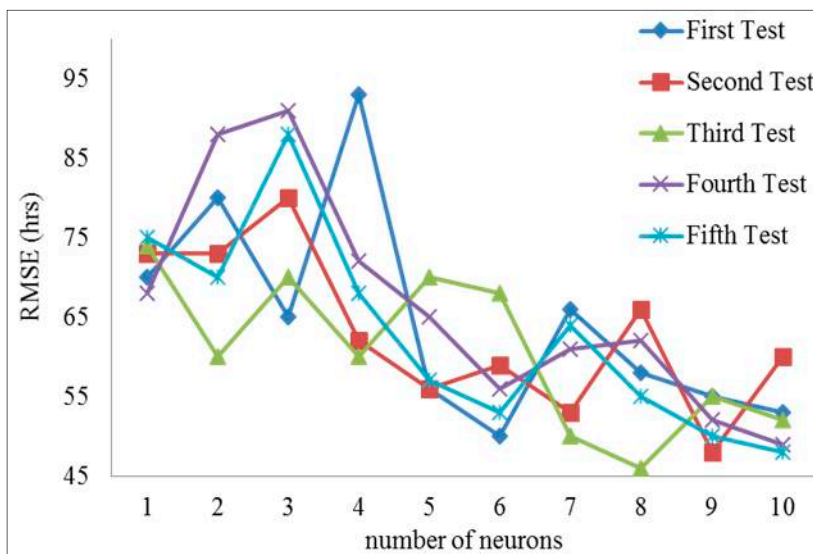


Figure 8. Effects of the number of epochs.

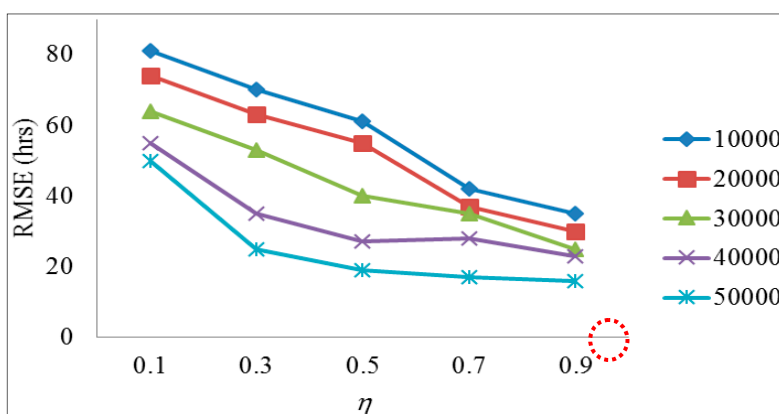


Table 4. The estimation accuracy of the PCA-CART-BPN method (four-fold cross-validation).

	MAE (h)	MAPE	RMSE (h)
First Run	32.8	2.8%	51.7
Second Run	30.2	2.7%	48.5
Third Run	37.0	3.3%	54.5
Fourth Run	35.8	3.2%	54.5
Average	34.0	3.0%	52.3

Table 5. The estimation accuracy of the CART method (four-fold cross-validation).

	MAE (h)	MAPE	RMSE (h)
First Run	115	9.2%	100
Second Run	90	7.4%	87
Third Run	116	9.6%	99
Fourth Run	132	11.0%	109
Average	113	9.3%	99

**Table 6.** The estimation accuracy of the CART-BPN method (four-fold cross-validation).

	MAE (h)	MAPE	RMSE (h)
First Run	90.1	7.3%	89.3
Second Run	74.8	5.7%	76.0
Third Run	41.0	3.4%	55.3
Fourth Run	77.1	6.2%	82.1
Average	90.2	7.2%	89.5

**Table 7.** The estimation accuracy of the PCA-BPN method (four-fold cross-validation).

	MAE (h)	MAPE	RMSE (h)
First Run	41.7	3.7%	60.3
Second Run	38.6	3.4%	55.8
Third Run	40.8	3.5%	59.9
Fourth Run	33.9	2.9%	53.6
Average	38.7	3.4%	57.4

To evaluate which method among the multiple linear regression (MLR), BPN, CART, CBR, CART-BPN, PCA-BPN, kM-BPN, and PCA-CART-BPN methods was more accurate, the estimation performance of the eight methods was arranged, as shown in Table 8, and the RMSE, MAE, and MAPE were employed to determine estimation accuracy. However, we did not compare with methods based on expert collaboration, such as Chen [7] and Chen [10], because there is no collaboration part in the proposed methodology. In addition, some hybrid methods including kM-FBPN [15], FCM-BPN and SOM-FBPN [12] are similar in nature to kM-BPN, and therefore were not compared. Further, the symmetric-partitioning and incremental-relearning classification and BPN approach proposed by Chen [18] was not compared because the execution time was more than 5 min. Only the estimation performances to the testing/validation data were compared.

**Table 8.** Comparisons of the performances of various methods.

Performance Measure	MLR	BPN	CART	CBR	CART-BPN	PCA-BPN	kM-BPN	PCA-CART-BPN
RMSE (h)	89.5	144	132	168	75.7	57.4	119	52.3
MAE (h)	90.2	117	113	140	70.7	38.7	99	34.0
MAPE	7.2%	9.1%	8.9%	11.2%	5.6%	3.4%	7.7%	3.0%

- (1) As evident from Tables 5 and 6, the most obvious advantage of CART-BPN over CART was regarding MAPE, which achieved 10%. The CART-BPN prediction error was smaller, whereas the CART prediction error was larger; thus, the CART-BPN predictions were more accurate. To validate the more accurate prediction of the CART-BPN, the paired  $t$  test was employed.

$H_0$ : When estimating the job cycle time, the estimation accuracy of the CART-BPN method is the same as that obtained from using the CART method.

$H_1$ : When estimating the job cycle time, the CART-BPN method is more accurate than the CART method.



As expected, the CART-BPN forecasting method was superior to the CART method. Table 9 summarizes the results of a comparative analysis. Therefore, a nonlinear analysis is indeed beneficial to the job cycle time estimation problem.

**Table 9.** CART-BPN and CART of the paired  $t$  test.

	CART-BPN	CART
Mean	117.32	83.8
Variance	6779.72	3928.02
Number of observations	30	30
Pearson's Correlation Coefficient		0.2363
Degree of freedom		29
$t$ statistic		-2.0193
$P(T \leq t)$ one-sided		0.0263
$P(T \leq t)$ two-sided		0.0527

- (2) Of these, the PCA-CART-BPN estimation efficacy exceeded that of the statistical regression model (MLR) by 41%. To make sure that such a difference is statistically significant, a paired  $t$  test was used:

$H_0$ : When estimating the job cycle time, the PCA-CART-BPN methodology for estimating accuracy is the same as using the statistical regression approach.

$H_1$ : When estimating the job cycle time, the PCA-CART-BPN methodology is more accurate than the statistical regression approach.

The comparison results are summarized in Table 10. As a result, the advantage of the PCA-CART-BPN methodology over statistical regression was statistically significant at  $\alpha = 0.1$ .

**Table 10.** PCA-CART-BPN and regression of the paired  $t$  test.

	PCA-CART-BPN	MLR
Mean	94.53	37.32
Variance	2609.49	1554.66
Number of observations	30	30
Pearson's Correlation Coefficient		0.0287
Degree of freedom		29
$t$ statistic		-4.9247
$P(T \leq t)$ one-sided		0.0000156
$P(T \leq t)$ two-sided		0.0000312

- (3) When exploring the PCA-CART-BPN and grouping CART-BPN and kM-BPN models, utilizing the PCA-CART-BPN method reduced the MAE by 52% and 66%, respectively. Therefore, PCA-CART was a better job classifier than kM and CART.
- (4) The most obvious advantage of the proposed methodology over CART was up to 70% in terms of MAE.
- (5) The most obvious advantage of the proposed methodology over CART-BPN was up to 52% in terms of MAE, which confirmed the effectiveness of variable replacement using PCA.

- (6) If the differences between the attributes of two jobs could be fully reflected on their cycle times, CBR would be very effective. However, the results did not support this viewpoint.
- (7) The same problem happened to kM-BPN. Nevertheless, the performance of kM-BPN was quite close to that of the proposed methodology, which can be attributed to the approximation ability of BPN.
- (8) The execution times of the eight methods were compared in Table 11. Direct methods like MLR were really fast. Iterative methods like CART, CBR, and BPN were a bit slow. Nevertheless, training a BPN using the LM algorithm was also efficient. The execution times of some hybrid methods (CART-BPN, PCA-BPN, kM-BPN, and PCA-CART-BPN) were approximately equal, which supported the reasonability of using the proposed methodology instead of CART-BPN, PCA-BPN, or kM-BPN.

**Table 11.** The comparison results.

	Execution Time (s)
MLR	<1
BPN	16
CART	5
CBR	15
CART-BPN	127
PCA-BPN	130
kM-BPN	132
PCA-CART-BPN	132

#### 4. Managerial Implications

Estimating and shortening the cycle time of each job is an important task to maintain a competitive edge in the DRAM industry. For example, the famous DRAM maker, Samsung, implemented the short cycle time and low inventory (SLIM) method to estimate the cycle times and WIP levels for various manufacturing steps, so that a more effective control of the factory was possible. As a result, the average cycle times of some DRAM products were reduced from more than 80 days to less than 30 days, bringing Samsung a benefit of about 1 billion US\$ [23].

Incorrectly estimating the cycle times of jobs increases the difficulties in projecting the monthly output of a wafer fabrication factory [10], which then misleads the production planning personnel in making the release plan. If many wafers are incorrectly outputted from a wafer fabrication factory during a month with very low average selling prices (ASPs), the wafer fabrication factory will suffer considerable losses. To illustrate this, an example is given as follows. There are two products, A and B, in the factory. The ASPs of the two products are shown in Table 12. The gross dies and wafer yields of the two products are the same: 500 dies per wafer and 95%. The monthly capacity of the factory is 10,000 pieces of wafers distributed between the two products according to their ASPs. The actual cycle times of the two products are two and three months, respectively. The release plan based on the correct cycle times is shown in Table 13. The yearly revenues (months 1 to 12) are 205 million US\$. If the cycle time of product A is mistaken as three months, the release plan will be Table 14. The yearly revenues reduce to 203 million US\$, resulting in a loss of two million dollars per year.

**Table 12.** The average selling prices (ASPs) of the two products.

<b>Month</b>	<b>ASP (US\$) (Product A)</b>	<b>ASP (US\$) (Product B)</b>
-10	4.8	3.5
-11	3.4	3.4
-12	2.5	3.5
1	3.9	3.5
2	4.2	3.6
3	2.3	3.5
4	3.9	3.4
5	4.5	3.5
6	4.4	3.5
7	2.6	3.3
8	2.2	3.5
9	4.0	3.5
10	2.6	3.6
11	2.8	3.5
12	4.7	3.4
+1	2.7	3.5
+2	3.5	3.5
+3	4.2	3.3

**Table 13.** The correct release plan.

<b>Month</b>	<b>Release (Product A)</b>	<b>Release (Product B)</b>
-10	4139	5861
-11	5236	4764
-12	5441	4559
1	3965	6035
2	5285	4715
3	5605	4395
4	5536	4464
5	4290	5710
6	3860	6140
7	5308	4692
8	4253	5747
9	4396	5604
+10	5703	4297
+11	4379	5621
+12	4991	5009

**Table 14.** The incorrect release plan.

Month	Release (Product A)	Release (Product B)
−10	5236	4764
−11	5441	4559
−12	3965	6035
1	5285	4715
2	5605	4395
3	5536	4464
4	4290	5710
5	3860	6140
6	5308	4692
7	4253	5747
8	4396	5604
9	5703	4297
+10	4379	5621
+11	4991	5009
+12	5422	4578

## 5. Conclusions and Future Research Directions

Cycle time reduction continues to be one of the major concerns in controlling a wafer fabrication factory. On one hand, long cycle times lead to the accumulation in WIP. On the other hand, the risk of wafer contamination also increases when the cycle time of a job is long. Cycle time estimation, which is considered as a prerequisite for achieving that, has received much attention.

In this study, the PCA-CART-BPN approach is proposed to estimate the cycle time of a job in a wafer fabrication factory. First, PCA is applied to construct a series of linear combinations of original variables to form new variables that are as unrelated to each other as possible. According to the new variables, jobs are classified using CART before estimating their cycle times with BPNs. The advantage of such a treatment is the consideration of the estimation performance when classifying the jobs. To the contrary, in the existing methods such as kM-BPN, SOM-FBPN, and others, job classification only considers the differences between the attributes of jobs.

After evaluating the estimation performance of the proposed PCA-CART-BPN approach with a real case:

- (1) The advantage of the proposed PCA-CART-BPN approach over the existing methods was quite obvious. Three measures of the estimation accuracy, especially MAE, were improved after applying PCA-CART-BPN.
- (2) In addition, PCA, CART, and BPN are all mature methods supported by a lot of optimization packages (see Table 15), which is conducive to the joint use of the two methods.
- (3) On the other hand, the efficiency of PCA-CART-BPN was not poorer than that of the existing hybrid methods like CART-BPN, PCA-BPN, and kM-BPN.

**Table 15.** Some software packages for implementing PCA, CART, and BPN.

Method	Software
PCA	MATLAB, SAS, SPSS, Stata
CART	CART (Salford Systems), MATLAB, SAS
BPN	Brian, GENESIS, MATLAB (Neural Network Toolbox), NEURON

In future studies, a collaboration mechanism can be incorporated into the PCA-CART-BPN approach, so that multiple experts can estimate the cycle time of a job collaboratively.

### Acknowledgment

This work was supported by National Science Council of Taiwan.

### Author Contributions

Toly Chen proposed the PCA-CART-BPN approach, collected the data, and wrote most of the content of this paper. Yu-Cheng Wang analyzed the data, prepared the figures of the results, and helped revise the paper.

### Conflicts of Interest

The authors declare no conflict of interest.

### References

1. Leachman, R.C.; Hodges, D.A. Benchmarking semiconductor manufacturing. *IEEE Trans. Semicond. Manuf.* **1996**, *9*, 158–169.
2. Inoue, T.; Ishii, Y.; Igarashi, K.; Takagi, H.; Muneta, T.; Imaoka, K. Conveyer Belt Model to Analyze Cycle Time Conditions in a Semiconductor Manufacturing Line. In Proceedings of the IEEE International Symposium on Semiconductor Manufacturing Conference, Tokyo, Japan, 25–27 September 2006; pp. 74–77.
3. Wein, L.M. On the relationship between yield and cycle time in semiconductor wafer fabrication. *IEEE Trans. Semicond. Manuf.* **1992**, *5*, 156–158.
4. Chen, T. A systematic cycle time reduction procedure for enhancing the competitiveness and sustainability of a semiconductor manufacturer. *Sustainability* **2013**, *5*, 4637–4652.
5. Chen, H.; Harrison, J.M.; Mandelbaum, A.; van Ackere, A.; Wein, L.M. Empirical evaluation of a queueing network model for semiconductor wafer fabrication. *Oper. Res.* **1988**, *36*, 202–215.
6. Pearn, W.L.; Chung, S.L.; Lai, C.M. Due-date assignment for wafer fabrication under demand variate environment. *IEEE Trans. Semicond. Manuf.* **2007**, *20*, 165–175.
7. Chen, T. An effective fuzzy collaborative forecasting approach for predicting the job cycle time in wafer fabrication. *Comput. Ind. Eng.* **2013**, *66*, 834–848.
8. Wu, N.; Chu, F.; Chu, C.; Zhou, M. Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting. *IEEE Trans. Syst. Man Cybernet. Syst.* **2013**, *43*, 196–207.

9. 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.
10. Chen, T. A fuzzy-neural knowledge-based system for job completion time prediction and internal due date assignment in a wafer fabrication plant. *Int. J. Syst. Sci.* **2009**, *40*, 889–902.
11. 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.
12. Chen, T. A hybrid look-ahead SOM-FBPN and FIR system for wafer lot output time prediction and achievability evaluation. *Int. J. Adv. Manuf. Technol.* **2007**, *35*, 575–586.
13. Chen, T. A fuzzy back propagation network for output time prediction in a wafer fab. *Appl. Soft Comput.* **2003**, *2*, 211–222.
14. 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.
15. Chen, T. An intelligent hybrid system for wafer lot output time prediction. *Adv. Eng. Inf.* **2007**, *21*, 55–65.
16. Chen, T.; Wang, Y.C. An iterative procedure for optimizing the performance of the fuzzy-neural job cycle time estimation approach in a wafer fabrication factory. *Math. Probl. Eng.* **2013**, *2013*, 1–15.
17. Chen, T. A hybrid SOM-BPN approach to lot output time prediction in a wafer fab. *Neur. Process. Lett.* **2006**, *24*, 271–288.
18. Chen, T. The symmetric-partitioning and incremental-relearning classification and back-propagation-network tree approach for cycle time estimation in wafer fabrication. *Symmetry* **2014**, *6*, 409–426.
19. Breiman, I.; Friedman, J.H.; Olsen, R.A.; Stone, C.J. *Classification and Regression Trees*; Wadsworth: Pacific Grove, CA, USA, 1984.
20. Steinburg, D.; Colla, P. *Classification and Regression Trees*; Salford Systems: San Diego, CA, USA, 1997.
21. Firoze, A.; Arifin, M.S.; Rahman, R.M. Bangla user adaptive word Speech recognition: approaches and comparisons. *Int. J. Fuzzy Syst. Appl.* **2013**, *3*, 1–36.
22. Rogers, L.L.; Dowla, F.U. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. *Water Resour. Res.* **1994**, *30*, 457–481.
23. Ranganathan, A. The Levenberg-Marquardt Algorithm. Available online: <http://www.scribd.com/doc/10093320/Levenberg-Marquardt-Algorithm> (accessed on 28 February 2014).
24. Leachman, R.C.; Kang, J.; Lin, V. SLIM: Short cycle time and low inventory in manufacturing at Samsung Electronics. *Interfaces* **2002**, *32*, 61–77.
25. Chang, P.-C.; Hsieh, J.-C.; Liao, T.W. Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory. *J. Intell. Manuf.* **2005**, *16*, 549–557.
26. Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C. Genetic programming for evolving due-date assignment models in job shop environments. *Evol. Comput.* **2014**, *22*, 105–138.