International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): 810-820
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

Solving Small Sample Recipe Generation Problem with Hybrid WKRCF-PSO

Mohd Ibrahim Shapiai[1], Zuwairie Ibrahim[1], Marzuki Khalid[1], Lee Wen Jau[2], Soon-Chuan Ong[2] and Vladimir Pavlovich[3]
[1]Centre of Artificial Intelligent and Robotics (CAIRO), Universiti Teknologi Malaysia, Jalan Semarak, 54100, Kuala Lumpur, Malaysia.
[2]ATTD Automation (APAC) Pathfinding, Intel Technology Sdn. Bhd. Kulim, Penang, Malaysia
[3]Department of Computer Science, Rutgers University, NJ 08854, New Jersey, United States
ibrahimfke@gmail.com, zuwairiee@fke.utm.my, marzuki.khalid@utm.my, wen.jau.lee@intel.com, soon.chuan.ong@intel.com and vladimir@cs.rutgers

ABSTRACT

The cost of the experimental setup during the assembly process development of a chipset, particularly the under-fill process, can often result in insufficient data samples. In INTEL Malaysia, for example, the historical chipset data from an under-fill process consist of only a few samples. As a result, existing machine learning algorithms for predictive modeling cannot be applied in this setting. Despite this challenge, the use of data driven decisions remains critical for further optimization of this engineering process. In the proposed framework, the original weighted kernel regression with correlation factor (WKRCF) is strengthened by normalizing the input parameters and employing the Particle Swarm Optimization (PSO) as weight estimator. It is found that PSO gives flexibility in defining the objective function as compared to the iteration technique of WKRCF. Thus, an assumption on noise contamination to the available training samples can be implemented. Even though only four samples are used during the training stage of the conducted experiment, the proposed approach is able to provide better prediction within the engineer's requirements as compared with WKRCF. Thus, the proposed approach is beneficial for recipe generation in an assembly process development.

KEYWORDS

Correlation Factor, Normalization, Particle Swarm Optimization, Recipe Generation, Small Samples, Weighted Kernel Regression

1 INTRODUCTION

Recipe generation provides the key references needed by engineers to set up a new experiment for a new product and plays an important role in determining the success of product development. Currently, the ingredients chosen for the recipe mainly depend on the engineer's knowledge. Optimizing the input parameters will facilitate the engineering decisions needed to fulfill certain requirements. As the assembly process for chipsets is rapidly progressing towards smaller scales and greater complexity, the accuracy and efficiency requirements are more vital. For example, a semiconductor process flow requires hundreds of fabrication operations steps with a lead-time of a few months. In addition, device fabrication and manufacturing costs

continue to escalate. In addition to the usual strategy of increasing the wafer size and shrinking devices to reduce the cost per transistor, automation and modeling are becoming more important. Fowler [1] revealed that the productivity improvement strategy of semiconductor manufacturing is based on operational improvement at the front-end of wafer fabrication; this strategy accounts for almost half of the total annual productivity improvement target.

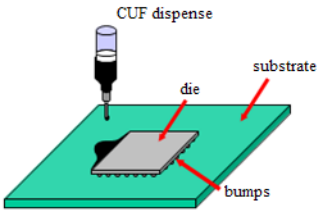

Figure 1. Illustration of an under-fill process in an assembly process.

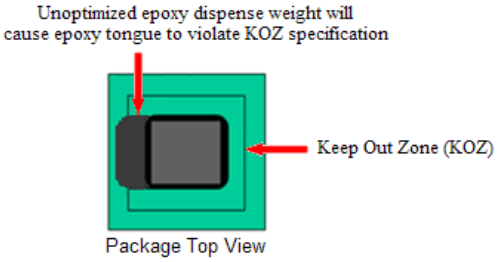

Figure 2. Illustration of an epoxy tongue that touch the keep out zone.

The use of artificial intelligence techniques for process modeling during the downstream assembly and all the involved tests is expected to reduce the overall manufacturing cost. As artificial intelligence techniques have been successfully applied in various engineering applications [2-3], introducing intelligent modeling to the assembly process promises to accelerate the engineering decisions even at early stages when very few collected samples are available. Inherently, intelligent modeling can improve equipment and resource utilization. In general, the development of recipe generation for assembly processes has only limited samples. However, most of the current machine learning algorithms are hindered by the limited number of available samples. In other words, the performance of existing algorithms degrades because the sample size is insufficient [4].

In INTEL Malaysia, the under-fill process shown in Figure 1, which consists of six input parameters with a small and sparse data set, is considered. Those input parameters are die size (dimension of die), gap height, the number of bumps, dispense distance, dispense weight, and the output is the dispense tongue length. In practice, it is difficult to define the input-output relationship, and improperly determined input setting parameters frequently cause the yield to be 'excess epoxy', 'epoxy on die', or 'insufficient epoxy'. Notably, the experiment usually involves large samples, and it is rather expensive to determine the recipe that prevents the tongue generated during the under-fill process from touching the keep out zone (KOZ), as illustrated in Figure 2. Hence, it is important to develop a cost-effective method to arrive at the optimal setting.

The problem being solved can be categorized as of learning from small samples which has gained increasing attention in many fields, such as in assembly process for sparse prediction modeling [5-6], engine control modeling [7], medical problem [8], and pulp and paper industry [9]. In general, most of the existing techniques rely on the pre-data processing technique, utilizing bias

data points, and artificial samples generation in solving the problem.

Previously, WKRCF has proved to solve small sample with good accuracy for recipe generation problem [5] by pre-processing the input parameters with respect to the calculated correlation factor. However, the selection of smoothing parameter, *h*, is not easy as it is subject to large value of the input parameters and the objective function is only limited to the closed-form solution problem. Therefore, the main objective of this study is to enhance WKRCF by normalizing the input parameters and introduced PSO as technique in estimating the weight parameters. Also, the study aims to provide more flexibility in modeling the recipe generation problem thus the established model can be employed later by engineers easily.

The remainder of this paper is organized as follows. A brief review of the WKRCF is given in Section 2. A review of PSO is given in section 3. The proposed technique is presented in Section 4. Section 5 includes the implementation of the proposed technique and the experimental results. Finally, the conclusions are provided in Section 6.

2 WEIGHTED KERNEL REGRESSION WITH CORRELATION FACTOR REVIEW

In this section, we first review the basic algorithm of the WKRCF .The concept of the WKRCF is introduced in the following. Given training samples, $\{x_i, y_i\}_{i=1}^{n}$, where n is the number of training samples, $x_i \in \Re^d$ is the input and $y_i \in \Re$ is the target output. WKRCF is the technique to regress the output space by mapping the input space $\Re^d$ to $\Re$. In general WKRCF is a modified Nadaraya-Watson kernel regression (NWKR) by expressing the weight based on the observed samples through a kernel function. The existing WKRCF relies on the Gaussian kernel function as given in Eq. (1) with subject to the correlation factor, c_p

$$K(X,X_i) = \frac{1}{\sqrt{2\pi}} \exp \frac{\left(-\|X - X_i\|^2\right)}{h \times c_p} \quad (1)$$

where h is the smoothing parameter. As in NWKR, the selection of smoothing parameter, *h*, is important to compromise between smoothness and fitness [6]. As in existing WKRCF, Eq. (2) is employed to determine the value of *h*.

$$h = \sum_{i=1}^{n} \left(\|X_i\|^2 - \overline{\|X\|}^2 \right)^2 \quad (2)$$

The correlation factor is introduced to adaptively set the smoothing parameter, *h*, of the Gaussian Kernel Function for WKRCF. Initially, the correlation coefficient for each input parameter must be calculated as follows

$$r_{x_p y} = \frac{\mathrm{cov}(x_i, y)}{\sigma_{x_p} \sigma_y} = \frac{E\{(x_p - \bar{x}_p)(y - \bar{y})\}}{\sigma_{x_p} \sigma_y} \quad (3)$$

where $r_{x_p y}$ is the correlation coefficient, x_p and $\bar{x}_p$ are the input value of one particular dimension and the corresponding mean value of the set of x_p respectively y and $\bar{y}$ are the output value and the corresponding mean value of the set of y, σ_{x_p} is the standard deviation of x_p and σ_y is the standard deviation of y.

Then, the correlation factor, c_p, for each input parameter is then can be defined as:

$$c_p = \frac{r_{x_p y}}{\sum_{k=1}^{d} r_{x_k y}} \tag{4}$$

The kernel matrix $K = [K_{ij}]$, where $i = j = 1,..., n$, with a generalised kernel matrix based on the adaptive Gaussian kernel, is given in Eq. (5). The matrix K transforms the linear observed samples to non-linear problems by mapping the data into a higher dimensional feature space.

$$K_{ij} = \begin{cases} \dfrac{\prod_{p=1}^{d} K\left(X_i^p, X_j^p, c_p\right)}{\sum_{l=1}^{n}\left[\prod_{p=1}^{d} K\left(X_{i\vee j}^p, X_j^p, c_p\right)\right]} & i \neq j \\ \dfrac{1}{\sum_{l=1}^{n}\left[\prod_{p=1}^{d} K\left(X_{i\vee j}^p, X_j^p, c_p\right)\right]} & i = j \end{cases} \tag{5}$$

In WKRCF, the most popular function for regression problems is used which to minimize the sum of squared error (SSE) to estimate the weight parameters, W.

$$\min f(W) \Leftrightarrow \min \|Kw - y\|^2 \tag{6}$$

Once the optimum weight is estimated, the model is ready to predict any unseen samples (test samples). The test samples can be predicted by using Eq. (7)

$$\hat{y}\left(X, \hat{W}\right) = \frac{\sum_{i=1}^{n} \hat{w}_i \left(\prod_{p=1}^{d} K\left(X^p, X_i^p\right)\right)}{\sum_{i=1}^{n}\left(\prod_{p=1}^{d} K\left(X^p, X_i^p\right)\right)} \tag{7}$$

3 OVERVIEW OF PARTICLE SWARM OPTIMIZATION

PSO is originally proposed by James Kennedy and Russell C. Eberhart [7] which is inspired by the social behavior of birds in nature. This population-based search algorithm initialized the population which called particles in problem space by random. Each particle flies in the problem space looking for the optimal solution according to its own and its companion's flying experience. PSO is found to be a very popular choice to solve optimization problem which the decision variables are real number [8], easy to implement and computational efficient [9].

In PSO algorithm, every particle represents the possible solution in the problem space. As mentioned earlier, each particle will fly over *d*-dimensional problem space for searching the optimum solution by updating its own velocity, $v_{i,d}(t)$, and position, $p_{i,d}(t)$ with respect to the fitness function . The current velocity of each particle is updated based on the personal best previous position found so far by every i^{th} particle, $p_{i,d}^{pbest}$, and the global best previous position found so far by the swarm, p_d^{gbest}. The d^{th} dimensional of the velocity and position for i^{th} particle is updated using Eq. (8) and Eq. (9) respectively.

$$\begin{aligned} v_{i,d}(t+1) &= k v_{i,d}(t) + ... \\ &... c_1 r_1 \left(p_{i,d}^{pbest} - p_{i,d}(t)\right) ... \\ &... + c_2 r_2 \left(p_d^{gbest} - p_{i,d}(t)\right) \end{aligned} \tag{8}$$

$$p_{i,d}(t+1) = p_{i,d}(t) + v_{i,d}(t+1) \tag{9}$$

where t is the iteration value, c_1 and c_2 are the cognitive and social coefficients, r_1 and r_2 are random values in the range [0, 1] and k is the inertia weight. The

cognitive and social coefficients control the tendency of particles to move toward its own or the entire particles position.

The random values provide randomness exploitation for particle in the problem space. Meanwhile, the inertia weight controls the exploration of particle in finding the optimum solution. Large inertia weight cause larger exploration of the problem space, while smaller inertia weight focuses the search in a smaller region. In PSO, the inertia weight is decreased overtime with typically large initial value and the equation is given below

$$k = k_{init} - \frac{k_{init} - k_{final}}{iteration} \times iteration_t \quad (10)$$

where k_{init} and k_{final} are the predefined initial and final value of the inertia weight respectively, *iteration* is the maximum number of iteration while $iteration_t$ is a current iteration.

4 THE PROPOSED WKRCF - PSO

An overview of the proposed technique is given in Figure 3. The proposed technique requires a series of steps to develop the prediction model. As we mentioned before, the existing WKRCF does not require any pre-processing on the available dataset and the estimation technique is based on the iteration technique.

In this section, we highlight all the employed features in constructing the WKR-PSO. Initially, the historical Intel dataset is subjected to the simple normalization as to avoid the difficulty in selecting the smoothing parameter, *h*.

The normalization of the given input parameters is given in Eq. (11)

$$X_{i,p}^{norm} = \frac{X_{i,p}}{\max\left(X_p\right)} \quad (11)$$

where $X_{i,p}^{norm}$ is a normalized input parameter, $X_{i,p}$ is the original input parameter and $\max\left(X_p\right)$ is the maximum value of one particular dimension of the input parameter for all available training samples. Once all the input parameters are normalized, the smoothing parameter can be simply

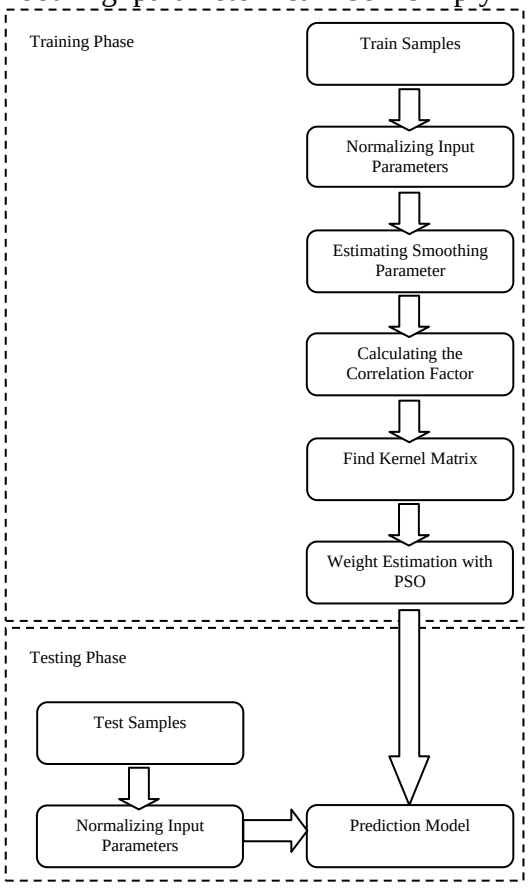

Figure 3. Overview of the proposed technique, WKRCF-PSO.

calculated based on Eq. (12). Initially, all the input parameters are arranged in ascending order of L_2-norm values.

$$h = \max\left(\left\| X_{k+1}^{norm} \right\|^2 - \left\| X_{k}^{norm} \right\|^2 \right) ... \quad ...where \quad 1 < k < n\text{-}1 \tag{12}$$

The correlation factor and the kernel matrix are calculated based on Eq. (4) and Eq. (5) respectively. In WKRCF [5], the weight parameters are estimated based on the iteration technique and subject to Eq. (6) . However, the existing WKRCF is only capable to solve closed form solution problem with a good accuracy. As PSO has the capability in solving non-closed form solution problem, the fitness function of the proposed technique is defined as in Eq. (13) by assuming that the available dataset set is imperceptibly corrupted by noise.

$$\min f_{fitness}(W) \Leftrightarrow \min \left\| Kw - y \right\|^2 + \lambda \left\| w \right\|_1 \tag{13}$$

where λ is a positive constant value, K is the kernel matrix, y is the target output and w is the weight parameters to be estimated. The λ value is usually determined based on the knowledge of the corrupted noise. In PSO, there are few parameters to be predefined before executing the algorithm. As the problem space corresponds to the continuous value of the estimated weight parameters, the selection of k_{init} and k_{final} values is more crucial. In this study, the value of k_{init} is purposely chosen to be large and the k_{final} value is chosen not to be very small as each particle is allowed to explore in wider problem space. This is to ensure the found solution represents the mostly likely solution.

Table 1. The normalized historical DOE data set; x and y is dimension size, gh is gap height, nb is number of bumps, dd is distance dispense, sw is amount of epoxy and output is length of tongue.

x	y	gh	nb	dd	sw	output
0.8583	0.7817	0.9044	1.0000	0.8462	1.0000	256305.3
1.0000	1.0000	0.9338	1.0000	1.0000	0.8000	166709.3
0.3579	0.3579	0.9154	0.0977	0.6154	0.2800	114980.7
0.9671	0.9492	1.0000	0.9460	0.9231	1.0000	250800.1
0.8583	0.7817	0.9044	1.0000	0.8462	0.8000	237581.9
1.0000	1.0000	0.9338	1.0000	1.0000	0.9000	243672.4
0.9671	0.9492	1.0000	0.9460	0.9231	0.8000	215971.4
0.9671	0.9492	1.0000	0.9460	0.8462	1.0000	246692.0
0.8583	0.7817	0.9044	1.0000	0.8462	0.6600	199574.8
0.8583	0.7817	0.9044	1.0000	0.8462	0.9400	251815.5

Table 2. Parameter settings for each of the function approximation algorithms

Technique	Parameter Settings
WKRCF	$h = \sum_{i=1}^{n} \left(\|X_i\|^2 - \overline{\|X\|^2} \right)^2$, iteration = 1000 (whichever is reached first)
WKRCF-PSO	$h = \max\left(\left\|X_{k+1}^{norm}\right\|^2 - \left\|X_{k}^{norm}\right\|^2 \right)$ *where* $1 < k < n\text{-}1$, Swarm Size = 100, Iteration = 500, $c_1 = c_2 = 1.4$, $k_{init} = 2.5$, $k_{final} = 0.4$ and $\lambda = 0.0001$

Once the weight parameters are estimated, the test samples have to be normalized subject to $\max(X_p)$ before translating into kernel space. Finally, all test samples can be predicted using Eq. (7).

4 EXPERIMENT AND RESULTS

4.1 Experiment Setup

In this investigation, the historical data set obtained from INTEL Malaysia [10] is employed in the experiment. Firstly the dataset is normalized by using Eq. (11) and it is shown in Table 1. The total number of available samples is ten and only four samples are used as training samples. The first four rows are chosen since those training samples cover the minimum and the maximum range of the input and output values. This is a relevant assumption as the problem becomes an interpolation problem based on the observed samples. The remaining samples are then used to measure the performance of the proposed model.

Initially, all the parameter settings for each predictive modeling algorithm are predefined. The parameter settings are summarised in Table 2.

4.2 Performance Measure

A simple but useful concept from [11] is used to evaluate the performance of the prediction based on the error of the acceptance rate, *E*, as given in Eq. (14)

$$E_i = \left| \frac{predict_i - actual_i}{predict_i} \right| \times 100\% \qquad (14)$$

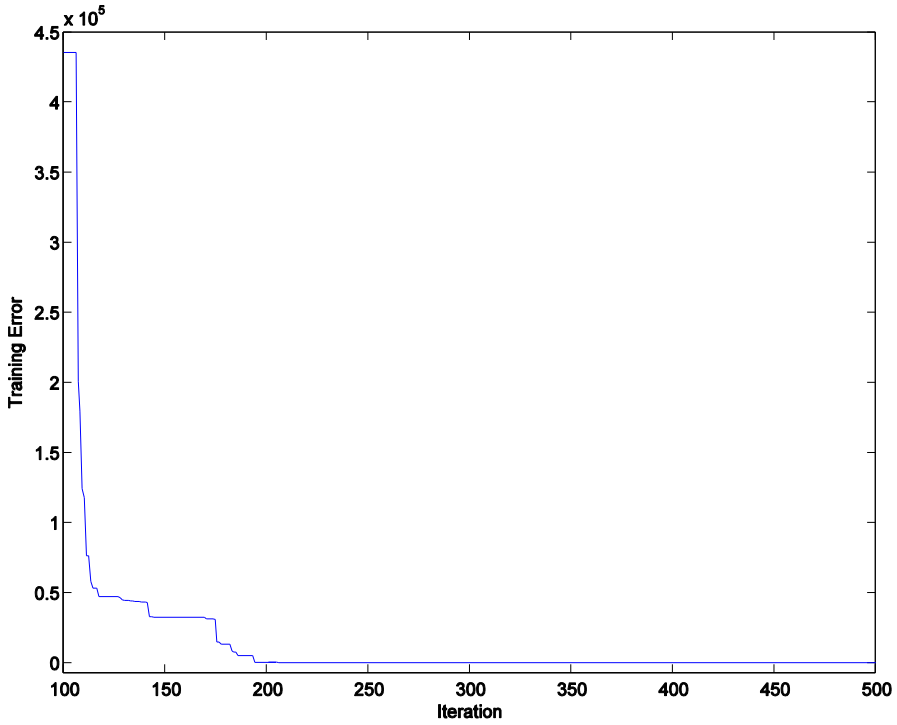

Figure 4. Convergence property of the proposed algorithm for the given problem from 100 iterations to the final iteration.

The acceptance rate is bounded by the guard band, *B*, as given in Eq. (15) in order to evaluate the coverage accuracy, *C*

$$E \le B \tag{15}$$

The coverage accuracy determines the quality of the prediction with respect to the predefined B value as given in Eq. (16)

$$C = \frac{total\ number\ of\ accepts}{total\ number\ of\ predicts} \times 100\% \tag{16}$$

In this study, we introduce new performance criterion, sum of error, *S*, to compare the prediction quality of the proposed technique with the existing techniques as given in Eq. (17)

$$S = \sum_{i=1}^{n} E_i \tag{17}$$

4.3 Results

The proposed technique is successfully trained by using PSO in estimating the weight parameters. Normalizing the input parameters avoids the difficulty in selecting smoothing parameter, *h*, and hence facilitates the training stage. In this study, the training is executed based on the predefined iteration where the algorithm reaches the convergence state as shown in Figure 4. Introducing the inertia weight, *k*, by selecting an appropriate k_{init} and k_{final} is important in estimating the weight parameters which are real number for high-dimensional problems. The selected k_{init} and k_{final} values are found to be the best combination for the given problem.

The presented results in Table 3 show the coverage accuracy, *C*, and the sum of error, *S*, of the two techniques for three

different guard band values. The WKR-PSO achieves better quality of prediction as compared to the existing WKRCF based on the calculated sum of error. Implicitly, WKRCF-PSO is successfully lower down the average of acceptance rate error for the entire prediction. However, the coverage accuracy of the proposed technique is equivalent to the WKRCF.

Table 3. The coverage accuracy of the presented techniques

Technique	Coverage Accuracy, *C* (%)			Sum of Error, *S*
	B=8%	B=12%	B=15%	
WKRCF	50	100	100	46.38
WKRCF-PSO	50	100	100	38.15

Table 4. Calculated correlation factor of the Intel Dataset

	c_1	c_2	c_3	c_4	c_5	c_6
Correlation Factor	0.1762	0.1570	0.0955	0.1983	0.1364	0.2365

As in WKRCF, introducing the correlation factor signifies the particular input relationship against the output by contributing more weight in predicting the output. The proposed normalization technique does not change the input(s) and the output relationship of the dataset. Thus the calculated correlation factor remains the same as in WKRCF as shown in Table 4. In logical sense, it is agreed with the nature of the dataset where the length of the tongue from the under fill process is highly correlated with the amount of the dispensed epoxy. Explicitly, the calculated coverage accuracy also agrees with the assumption on the correlation factor.

Finally, the chosen guard band values provide an indicator to engineer to set up a new experiment for a new product at certain confidence of level. As a result, the conducted experiment for the under fill process will fully utilizes the resources and indirectly reduces the cost by referring the recipe from the proposed model.

5 CONCLUSIONS

Because of limited information, learning from small samples is extremely difficult, especially for the under-fill process of an assembly process. This study shows that the modified version of WKRCF, namely WKRCF-PSO is better than the existing technique. Introducing the normalization technique successfully avoids the difficulty in choosing smoothing parameter and alleviates the training process. The proposed normalization technique also does not change the dataset relationship and

hence the assumption on the dataset dependencies remains the same as in WKRCF. This assumption is important especially in the region where there are no training samples available. Employing PSO as the weight estimator allows the non-closed form solution fitness function to be chosen in estimating the weight parameters. Thus an assumption of weak noise contamination on the available training samples can be introduced to the proposed technique. In general, it shows the flexibility of PSO in estimating the weight parameters as compared to the iteration technique. In the future, a technique to incorporate prior knowledge will be investigated to improve the prediction of the model.

Acknowledgments. This work was financially supported by the UTM-INTEL Research Fund (Vote 73332) and the Ministry of Higher Education (MOHE) by the Fundamental Research Grant Scheme (FRGS) (Vote 78564).

6 REFERENCES

1. Fowler, J.W.: Modeling and Analysis of Semiconductor Manufacturing. Dagstuhl Seminar, (2002).
2. Kurban, M., Filik, U.B.: Next Day Load Forecasting Using Artificial Neural Network Models with Autoregression and Weighted Frequency Bin Blocks International Journal Of Innovative Computing, Information And Control 5, 889-898 (2009).
3. Wakasa, Y., Tanaka, K., Akashi, T., Nishimura, Y.: PSO-Based Simultaneous Tuning Method for PID Controllers and Dead-Zone Compensators and Its Application to Ultrasonic Motors. International Journal Of Innovative Computing, Information And Control 6, 4593–4604 (2010).
4. Shapiai, M.I., Ibrahim, Z., Khalid, M., Jau, L.W., Pavlovic, V.: A Non-Linear Function Approximation From Small Samples Based on Nadaraya-Watson Kernel Regression. In: 2nd International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN2010), pp. 28- 32. (2010).
5. Shapiai, M.I., Ibrahim, Z., Khalid, M., Jau, L.W., Ong, S.-C., Pavlovich, V.: Recipe Generation from Small Samples: Incorporating an Improved Weighted Kernel Regression with Correlation Factor. In: Second International Conference, ICSECS 2011, pp. 144-154. Springer, Heidelberg, (2011).
6. Zhang, J., Huang, X., Zhou, C.: An improved kernel regression method based on Taylor expansion. Applied Mathematics and Computation 193, 419-429 (2007).
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: International Conference on Neural Networks, pp. 1942-1948 IEEE, (1995).

8. Engelbrecht, A.P.: Computational Intelligence, An Introduction. John Wiley & Sons, England (2002).
9. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA (2001).
10. Shapiai, M.I., Ibrahim, Z., Khalid, M., Jau, L.W., Ong, S.C., Pavlovich, V.: Recipe generation from small samples by weighted kernel regression. In: International Conference on Modeling Simulation and Applied Optimization (ICMSAO), pp. 1-4. IEEE, (2011).
11. Lee, W., Ong, S.: Learning from small data sets to improve assembly semiconductor manufacturing processes. In: 2nd ICCAE 2010, pp. 50-54. (2010).