

Modelling and optimization of fluid dispensing for electronic packaging using neural fuzzy networks and genetic algorithms

K.Y. Chan^{1,2} and C.K. Kwong¹

¹Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong; Email: C.K.Kwong@inet.polyu.edu.hk

²Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia; Email: Kit.Chan@curtin.edu.au

Abstract

Determination of process conditions for a fluid dispensing process of microchip encapsulation is a highly skilled task, which is usually based on engineers' knowledge and intuitive sense acquired through long-term experience rather than on a theoretical and analytical approach. Facing with the global competition, the current trial-and-error approach is inadequate. Modelling the fluid dispensing process is important because it enables us to understand the process behaviour, as well as determine the optimum operating conditions of the process for a high yield, low cost and robust operation. In this research, modelling and optimization of fluid dispensing processes based on neural fuzzy networks and genetic algorithms are described. First, neural fuzzy networks approach is used to model fluid dispensing process for microchip encapsulation. An N-fold validation tests were conducted. Results of the tests indicate that the mean errors and variances of errors of the modeling based on the neural fuzzy networks approach are all better than those of the other existing approaches, statistical regression, fuzzy regression and neural networks, on modeling the fluid dispensing. It is then followed by the determination of process conditions of the process based on a genetic algorithm approach. Validation tests were

conducted. Results of them indicate that process conditions determined based on the proposed approaches can achieve the specified quality requirements.

Keywords: Fluid dispensing, microchip encapsulation, genetic algorithm, neural fuzzy networks

1. Introduction

In the competitive market of today, manufacturers need to control variability at each of the many processing steps in a manufacturing line. All the variables controlling the desired output in a given process need to be understood and optimized for tight control. To achieve them, it is necessary to develop an accurate model for describing the process. There are two major approaches of developing process models; analytical modelling and empirical modelling. Analytical models are established based on the physical understanding of the process and deploys the various physical laws, typically a set of governing partial differential equations. They are attractive since they provide a fundamental understanding of the relationships between the various input and output parameters. Quite a few analytical models were developed for various manufacturing processes such as fluid dispensing (Chen 2002, Li et al. 2001), injection molding (Chiang et al.1991) and transfer molding (Han et al. 2000). Because of the complex behaviour of fluid dispensing, high degree of uncertainty associated with the process and assumptions undertaken for developing analytical models, an analytical model for fluid dispensing, which can provide accurate results in real world environment, is not yet available.

As opposed to analytical models, empirical modelling is a popular approach to developing process models using experimental data, which is commonly welcomed by industries. Classical statistical regression method is a common empirical approach to develop process models (Seber 2003). However, if the behaviour of processes is vague or irregular, the obtained models have unnaturally too wide possibility range (Chen et al. 2004). As a result,

statistical regression models can be applied only if the given data are distributed according to a statistical model, and the relationship between dependent and independent variables is crisp. Artificial neural networks have been used to develop process models for various manufacturing processes such as resistance spot welding (Lin et al. 2007) and transfer molding (Tong et al. 2004). They have the capability to transform a nonlinear mathematical model into a simplified black-box structure. The advantages of using neural networks in process modelling are that they have learning and generalization abilities as well as nonlinearity. Previous research has already confirmed that a neural network is a powerful tool for modelling nonlinear, complex and noisy processes. However, previous studies also found that the performance of a developed neural network is quite dependent on the pre-defined neural network architectural design as well as on the setting of the neural network parameters. A fuzzy logic modelling technique has been applied successfully in developing models of various manufacturing processes such as the Flip-Chip bonding process (Kang et al. 1993), vapor phase soldering (Xie et al. 1995) and the waterjet depainting process (Babets & Geskin 2000). In this approach, the basic elements of a fuzzy logic model are the internal functions, the membership functions and the outputs. The use of several internal functions accounts for the fuzziness of the model. All of the statistical regression methods, neural networks and the fuzzy logic modeling approach normally require a large number of experimental data sets to develop models. Compared with these three approaches, fuzzy regression has a distinct advantage which is that an acceptable process model can be developed using fewer experimental data sets or/and incomplete data sets based on that approach. Limited research has been conducted so far towards the use of fuzzy regression in process modeling. An attempt has been made by Schaiable (Schaiable 1997) to model the vertical CVD process using the fuzzy regression method. Lai (Lai 1994) applied fuzzy regression to modeling the die casting process.

Recent research shows that the approach of neural fuzzy networks has been used to develop process models for various manufacturing processes which are both nonlinear and complex (Tzafestas & Zikidis 2001, Kang et al 1993, Brinksmeier et al. 1998, Petri et al. 1998, Ling et al. 2003). It has the capability to transform a nonlinear mathematical model into a simplified input-output structure (Ying 1998). Kang et al. (1993) have proved that the TSK fuzzy system approach outperforms statistical regression and polynomial models in both correlation and prediction in modelling of highly nonlinear systems. Compared with conventional approaches of fuzzy logic, recent research has shown that neural fuzzy systems can achieve better performance, at least in mathematical function approximation, compared with the conventional approaches with the same number of fuzzy sets used in input variables (Fiordaliso 2001).

After developing process models, process optimization can be carried out to determine the optimal/proper process parameters of manufacturing processes. Quite a few techniques for process optimization have been attempted in previous research such as response surface methodology (RSM), and genetic algorithms (GA). RSM has been proven to be effective in many applications of process optimization (Poon 2000, Xie & Lee 1994). Xie and Lee (1994) developed a fuzzy logic-RSM based process optimization algorithm with CVD process as an application. The constructed fuzzy logic model was employed to evaluate the gradient and a gradient descent method was then used to optimize the objective function. However, one of its limitations is that it usually requires a large number of experimental data sets, specially, in the nonlinear region. GAs have been applied in process optimization of various manufacturing processes successfully (Hussein & El-Ghazaly 2004, Tan & Yuen 2000, Wilson et al 2001). They outperform the classical gradient descent method that it is more likely to reach the global optimum in solving nonlinear problems (DeJong 1975, Goldberg 1989).

Fluid dispensing is the most popular process for performing the microchip encapsulation on chip-on-board (COB) packages. However, it is a highly non-linear process and creates a highly coupled multi-variable system that involves the extremely complex inter-relationships among the fluid properties, process conditions, needle design parameters and overall encapsulation quality. In semiconductor manufacturing, a trial-and-error method is still very common to be used to identify proper process parameters setting. However, this method involves long process development time and optimum encapsulation quality may not be obtained systematically. Empirical model of fluid dispensing process also involves cognitive uncertainties. For example, cognitive uncertainties exist in the process such as the human handling of substrate for inspection and measurement, incomplete/insufficient data sets for process modelling, operator-to-operator variability and human judgment on the surface quality of the encapsulation. Various approaches have been attempted to model fluid dispensing process including analytical approach (Chen 2002, Li et al. 2001), statistical regression (Kwong et al. 2007), artificial neural networks (Kwong et al. 2007) and fuzzy regression (Ip et al. 2003).

In this paper, neural fuzzy networks approach to develop process models of fluid dispensing for microchip encapsulation is described. The developed process models are used to formulate a multiobjective optimization problem, which is then solved by a genetic algorithm. By solving the optimization problem, an optimal setting of process parameters for the fluid dispensing can be obtained. Validation tests were performed to evaluate the proposed approach to modelling and optimization of fluid dispensing process. Results of the validation tests indicate that process parameter settings can be obtained based on the proposed approaches to achieve the specified quality requirements.

2. Fluid dispensing process

In this paper, fluid dispensing for microchip encapsulation was studied. In the fluid dispensing of microchip encapsulation, normally, silicon chips are covered using an X-Y numerically controlled dispensing system that delivers epoxy encapsulant through a needle. The material is commonly dispensed in a pattern, working from the center out. An fluid dam around the die site and second wire bond points can be made to contain the flow of material and make a more uniform looking part as shown in Figure 1. With assistance from the supporting company of this research, three significant process parameters and their normal operating ranges were identified as: The compressed air pressure (1 bar to 4 bar), x_1 ; the pump motor speed (400 rpm to 1000 rpm), x_2 ; and the height between the substrate and the needle (250 to 2000 steps of stepping motor), x_3 . Two quality characteristics were also identified as encapsulation weight (mg), y and encapsulation thickness (mm), z .

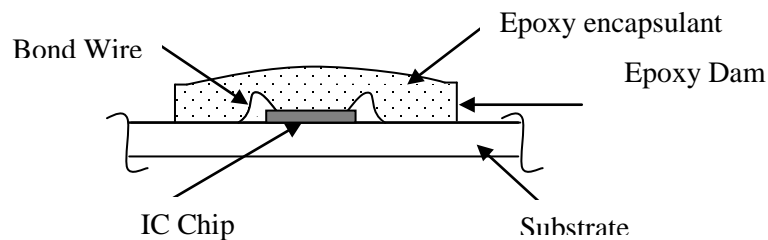


Figure 1 Encapsulation of COB packages

96 experiments were carried out based on a full factorial design with 4 levels in compressed air pressure (x_1), 6 levels in pump motor speed (x_2) and 4 levels in the height between the substrate and the needle (x_3). 88 out of the 96 experimental data were used to develop the neural fuzzy networks (NFN) based process model, and the rest 8 experimental data were used the validation test.

To evaluate prediction accuracy of the process model, the mean error (Me) was used as defined below:

$$\text{Me} = \frac{1}{n} \sum_{k=1}^n \frac{|t(k) - y(x(k))|}{t(k)} \quad (1)$$

where $t(k)$ is the actual value of the quality requirement for the k -th experimental data; $y(x(k))$ is the estimated quality requirement with the recommended process parameters $x(k)$ obtained from the process model; n is the number of experimental data sets used for the validation tests.

The variance of errors (Var) was also involved in the validation as shown below:

$$\text{Var} = \frac{1}{n-1} \sum_{k=1}^n \left(\frac{|t(k) - y(x(k))|}{t(k)} - \text{Me} \right)^2 \quad (2)$$

3. Modelling fluid dispensing using neural fuzzy network

The neural fuzzy network (NFN) approach is proposed to model the fluid dispensing process. Figure 2 shows the structure of a NFN which was used to model the encapsulation weight y and encapsulation thickness z . The network consists of an input layer in which an input vector containing process parameters x_1 , x_2 and x_3 , is fed in, the output layer which produces the output response, the quality characteristic y or z , and two hidden layers between the input and output layers. The functions of the four layers of the NFN are described below:

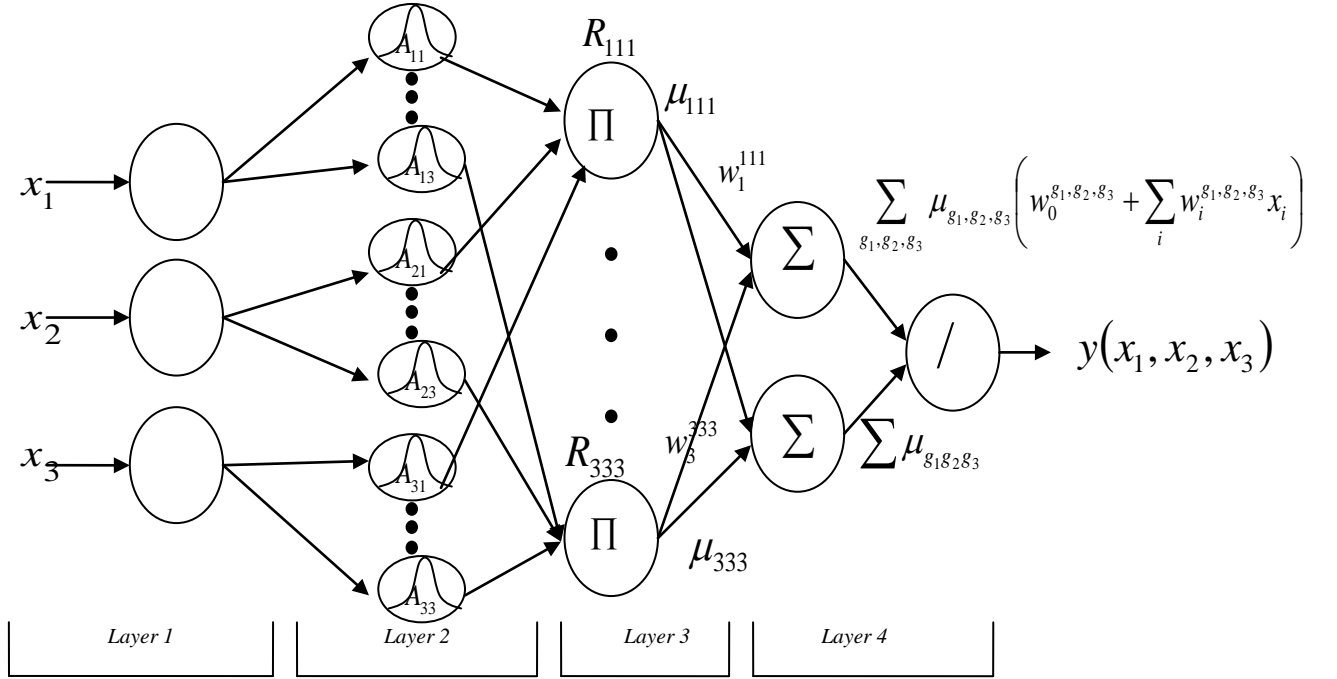


Figure 2 NFN based process model

Layer 1: No computation is done in this layer. Each node in this layer which corresponds to one process parameter transmits input values to the next layer.

Layer 2: Each node in this layer corresponds to one linguistic label (i.e. small, medium or large) of one of the input variables, x_1 , x_2 and x_3 , in Layer 1. Thus the number of fuzzy sets of each input variable is three. The membership value which specifies the degree to which an input value belongs to fuzzy set, is calculated in Layer 2. The j^{th} membership function of the input variable x_i is a bell-shaped function and is given by

$$A_{ij}(x_i) = e^{-\frac{(x_i - \bar{x}_{ij})^2}{2\sigma_{ij}^2}} \quad (3)$$

where parameters \bar{x}_{ij} and σ_{ij} with $i=1,2,3$ and $j=1,2,3$ are the mean value and the standard deviation of the j^{th} membership function of the input variable x_i respectively.

They are called NFN parameters which are searched by using a genetic algorithm.

The membership function $A_{ij}(x_i)$ of x_i with $j=1,2,3$ represent the small, medium and large fuzzy set respectively. Figure 3 shows the three membership functions of the process parameter x_1 . $A_{11}(x_1)$, $A_{12}(x_1)$ and $A_{13}(x_1)$ are the membership functions of x_1 for the fuzzy sets, ‘small’, ‘medium’ and ‘large’, respectively. \bar{x}_{11} , \bar{x}_{12} and \bar{x}_{13} are the mean values of the membership functions, $A_{11}(x_1)$, $A_{12}(x_1)$ and $A_{13}(x_1)$, respectively.

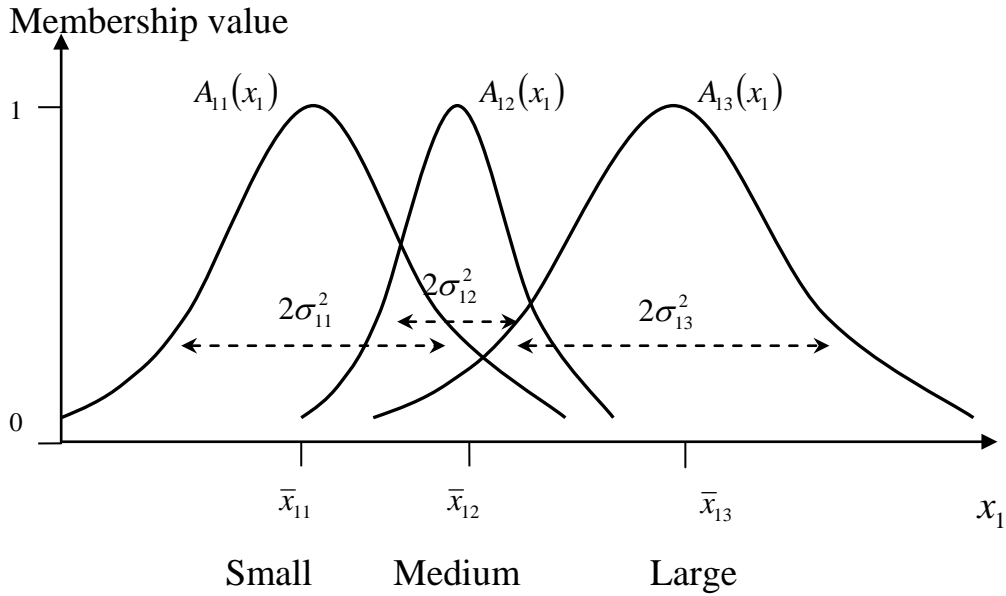


Figure 3 Membership functions for small, medium and large

Determination of proper setting of the neural fuzzy parameters \bar{x}_{ij} and σ_{ij} are important, because they directly affect the prediction accuracy of the neural fuzzy network based process models. In this research, the parameters were searched using Genetic Algorithm (GA) (Goldberg 1989). A discussion of using GA to search the neural fuzzy parameters is given in the end of this section.

Layer 3: A node in this layer represents one fuzzy logic rule and performs precondition matching

of a rule. For instance, a fuzzy rule $R_{g_1g_2g_3}$ can be given as:

$$R_{g_1g_2g_3} : \text{IF } x_1 \text{ is } A_{1g_1}(x_1) \text{ AND } x_2 \text{ is } A_{2g_2}(x_2) \text{ AND } x_3 \text{ is } A_{3g_3}(x_3) \text{ THEN } y(x_1, x_2, x_3) \text{ is } \mu_{g_1g_2g_3},$$

where $\mu_{g_1g_2g_3}$ is an output singleton y of the rule $R_{g_1g_2g_3}$, and it can be represented as:

$$\mu_{g_1g_2g_3} = A_{1g_1}(x_1) \cdot A_{2g_2}(x_2) \cdot A_{3g_3}(x_3) \quad (4)$$

In this project, number of membership functions of each input variable is 3 and the number of input variables is 3 as well, thus the total number of fuzzy rules is 27 (i.e. 3^3).

For example, fuzzy rule R_{213} can be written as:

R_{213} : IF x_1 is $A_{12}(x_1)$ AND x_2 is $A_{21}(x_2)$ AND x_3 is $A_{33}(x_3)$ THEN y is μ_{213} .

With $x_1 = 2.5$ bar, $x_2 = 500$ rpm and $x_3 = 1700$ steps, the membership values of x_1 , x_2 and x_3 can be found from Figure 4, 5 and 6 respectively, which are $A_{12}(x_1) = 0.5$, $A_{21}(x_2) = 0.1$ and $A_{33}(x_3) = 0.9$. Therefore the resulting output singleton y of the fuzzy rule R_{213} is:

$$\mu_{213} = A_{12}(x_1) \cdot A_{21}(x_2) \cdot A_{33}(x_3) = 0.045$$

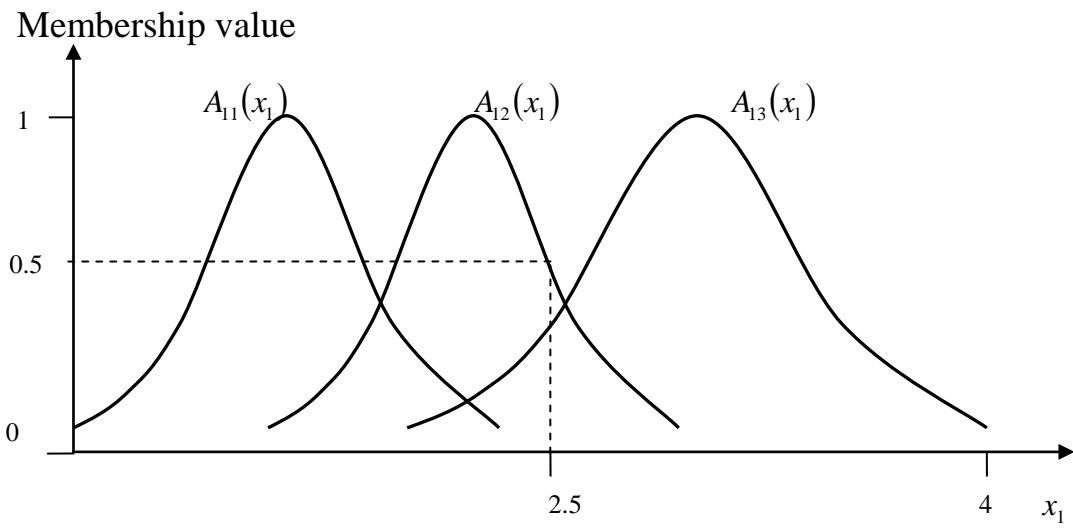


Figure 4 Membership functions of x_1

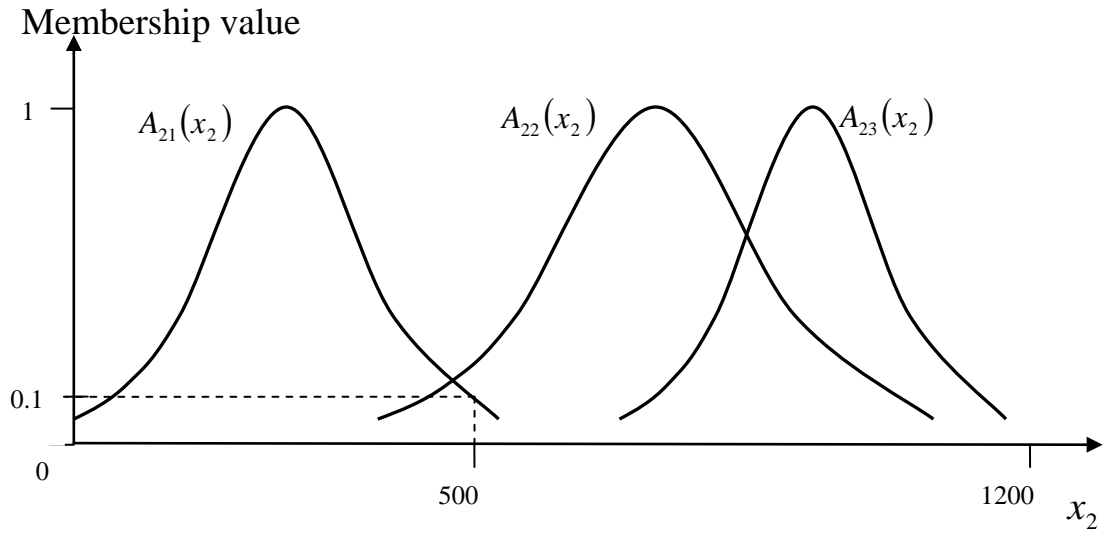


Figure 5 Membership functions of x_2

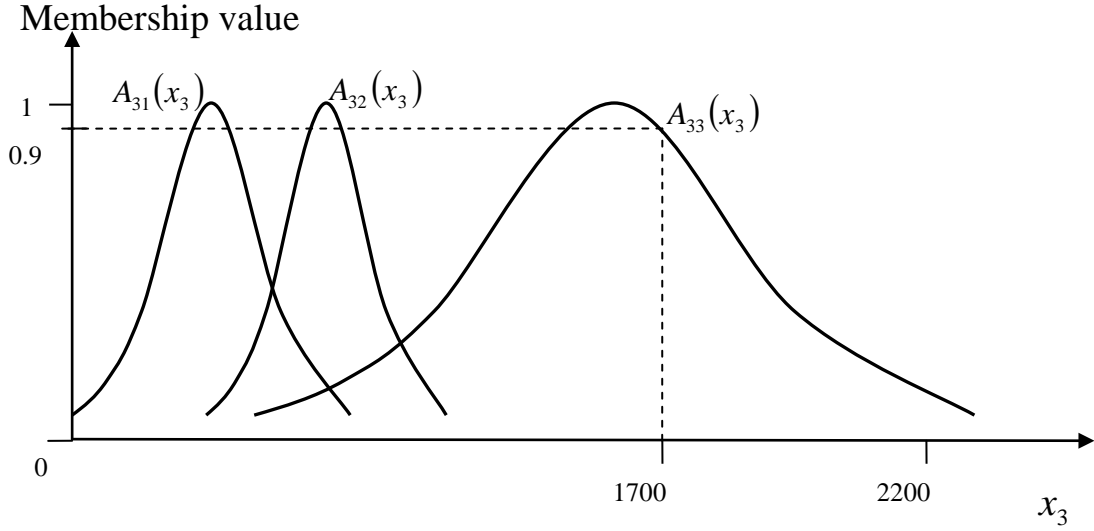


Figure 6 Membership functions of x_3

Layer 4: The node in this layer corresponds to the output variable y . The node integrates all the actions taken by the Layers 3 and acts as a defuzzifier with defined in (5).

$$y(x_1, x_2, x_3) = \frac{\sum_{g_1, g_2, g_3=1}^3 \mu_{g_1, g_2, g_3} (w_1^{g_1, g_2, g_3} x_1 + w_2^{g_1, g_2, g_3} x_2 + w_3^{g_1, g_2, g_3} x_3 + w_0^{g_1, g_2, g_3})}{\sum_{g_1, g_2, g_3=1}^3 \mu_{g_1, g_2, g_3}} \quad (5)$$

where the weights $w_0^{g_1, g_2, g_3}$, $w_1^{g_1, g_2, g_3}$, $w_2^{g_1, g_2, g_3}$ and $w_3^{g_1, g_2, g_3}$ with $g_1, g_2, g_3 = 1, 2, 3$ are the neural fuzzy parameters.

The NFN parameters including $w_0^{g_1, g_2, g_3}$, $w_1^{g_1, g_2, g_3}$, $w_2^{g_1, g_2, g_3}$ and $w_3^{g_1, g_2, g_3}$ where $g_1, g_2, g_3 = 1, 2, 3$, \bar{x}_{ij} and σ_{ij} with $i=1, 2, 3$ and $j=1, 2, 3$ were searched by using a genetic algorithm (Goldberg 1998). The objective of the genetic algorithm is to minimize the training error of the NFN model by searching the NFN parameters setting. The fitness function is defined as:

$$fitness = \frac{1}{1 + err} \quad (6)$$

where $err = \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} \frac{|y(\mathbf{x}) - y'(\mathbf{x})|}{y'(\mathbf{x})} \times 100\%$; $\mathbf{x} = (x_1, x_2, x_3)$, $y'(\mathbf{x})$ is the actual output, \mathbf{X} is the

domain of all the experimental data sets, and N is the total number of experimental data sets.

The range of *fitness* in (6) is $[0,1]$. With the use of GA, an optimal setting of neural fuzzy parameters can be obtained. The process starts with a population of randomly generated chromosomes, while each chromosome is set to be a set of neural fuzzy parameters. Thus the length of each chromosome is 45. Thereafter, the GA further processes a fixed number of generations using the evolutionary operations; selection, crossover and mutation. Selection is performed in two steps, chromosome selection and survival. Selection on chromosomes decides who become parents and how many children the parents have. Children are created through crossover which exchanges information between parents, and mutation which further perturbs the children. The fitness of each child is then evaluated based on the fitness function (6). Finally, the survival step decides who survives in the population. After the operations of selection, crossover and mutation, a new population can be generated. The chromosome with the highest fitness is selected to be the setting of the neural fuzzy parameters.

The following parameter values suggested by (Schaffer et al. 1989) have been used in the genetic algorithm for searching the neural fuzzy parameters; crossover rate = 0.8; mutation rate = 1/45, where 45 is the length of the chromosome; number of generations = 1000; population size = 200. The genetic algorithm for optimizing the NFN parameters was implemented using Matlab programming software.

Figure 7 and 8 show the searching results using genetic algorithm to develop the neural fuzzy models for the encapsulation weight and encapsulation thickness respectively.

A genetic algorithm run for developing the neural fuzzy model for encapsulation weight

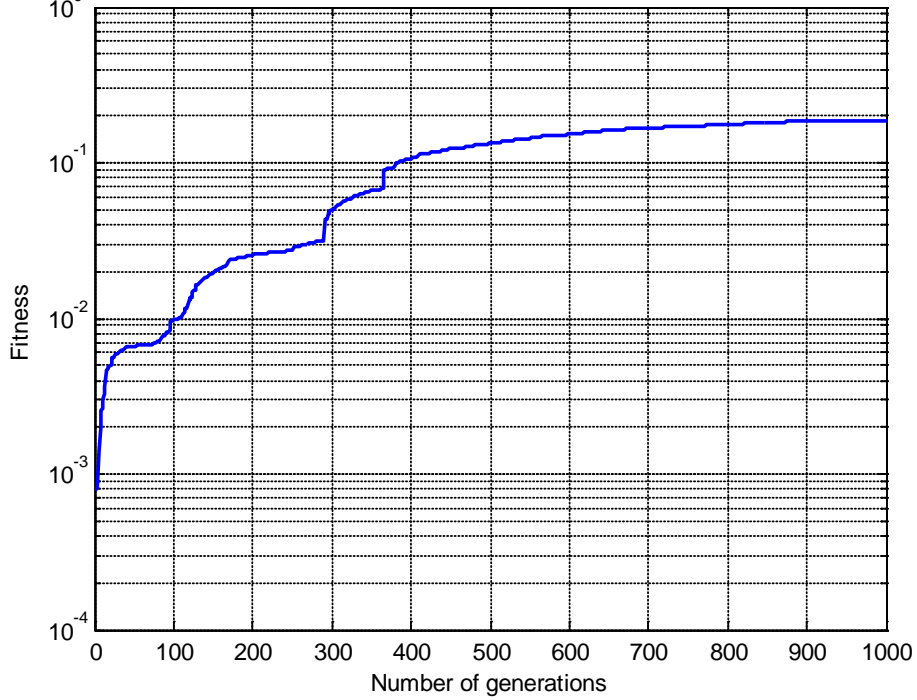


Figure 7 Searching results of developing the neural fuzzy model for encapsulation weight based on GA

A genetic algorithm run for developing the neural fuzzy model for encapsulation thickness

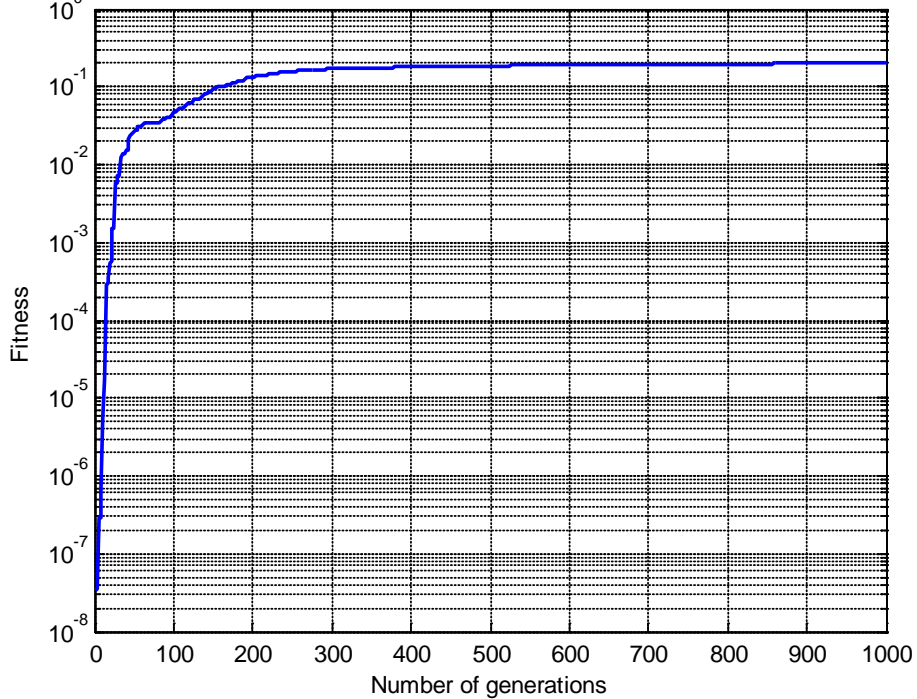


Figure 8 Searching results of developing the neural fuzzy model for encapsulation thickness based on GA

To validate the generalization ability of the developed NFN models in modelling the fluid dispensing process, N-fold validation tests were conducted. The validation results are compared with those based on statistical regression, artificial neural networks and fuzzy regression for modelling the fluid dispensing process (Kwong et al. 2007). 8 data sets were randomly selected from the 96 experimental data sets. The remaining 88 data sets are used for training the models. The validation test was repeated 11 times. For each validation test, a model was developed based on the 88 training data sets, and the remaining 8 data sets were used for validating the trained model. The mean errors were calculated based on (1) and the errors of variances were calculated based on (2). Table 1 and Table 2 in the appendix respectively summary the mean errors and the variances of errors of modelling the process with regard to encapsulation weight. Table 3 and Table 4 in the appendix respectively summarize the mean errors and the variances of errors of modelling the process with regard to the encapsulation thickness. From the tables, it can be seen that modelling the fluid dispensing process based on the NFN models yields the smallest mean errors and variances of errors. Results of the 11 validation tests are shown in Figure 9, 10, 11 and 12 in the appendix from which it can be seen that the prediction errors of NFN models, for all the tests, are the smallest.

Therefore the NFN based process models for encapsulation weight and thickness predictions were used to perform the process optimization of the fluid dispensing, and the genetic algorithm is proposed to determine an optimal setting of process parameters.

4. Process optimization using genetic algorithms

Given the required values, β mg and γ mm of the encapsulation weight and encapsulation thickness respectively, it is necessary to determine a setting of the process parameters for achieving the quality requirements of the encapsulation weight and thickness. To determine the process parameter setting, the following multi-objective functions are formulated by minimizing

the errors between the estimated values of quality characteristics and actual values of quality characteristics:

$$\text{Multi-objective function: } \begin{cases} \min f_1(x_1, x_2, x_3) = \frac{|y - \beta|}{\beta} \\ \min f_2(x_1, x_2, x_3) = \frac{|z - \gamma|}{\gamma} \end{cases} \quad (7)$$

subject to: {Operating range of process parameters (x_1 , x_2 and x_3)},

where the normal operating ranges of the three process parameters (x_1 , x_2 and x_3) are:

$1 \leq x_1 \leq 4$, $400 \leq x_2 \leq 1000$ and $250 \leq x_3 \leq 2000$. A set of process parameters of the multi-

objective optimization problem consists of all process parameter sets which cannot be improved

in any objective without degrading another objective. These process parameter sets are known as

Pareto-optimal. Mathematically, $\bar{x} = (x_1, x_2, x_3)$ dominates $\bar{x}' = (x_1', x_2', x_3')$ (also written as

$\bar{x} \succ \bar{x}'$) iff

$$\forall i \in \{1, 2\}: f_i(\bar{x}) \geq f_i(\bar{x}') \wedge \exists j \in \{1, 2\}: f_j(\bar{x}) > f_j(\bar{x}')$$

Additionally, \bar{x} covers \bar{x}' iff $\bar{x} \succ \bar{x}'$ or $\bar{x} = \bar{x}'$. All process parameter sets which are not

dominated by any other decision vector are called nondominated or Pareto-optimal.

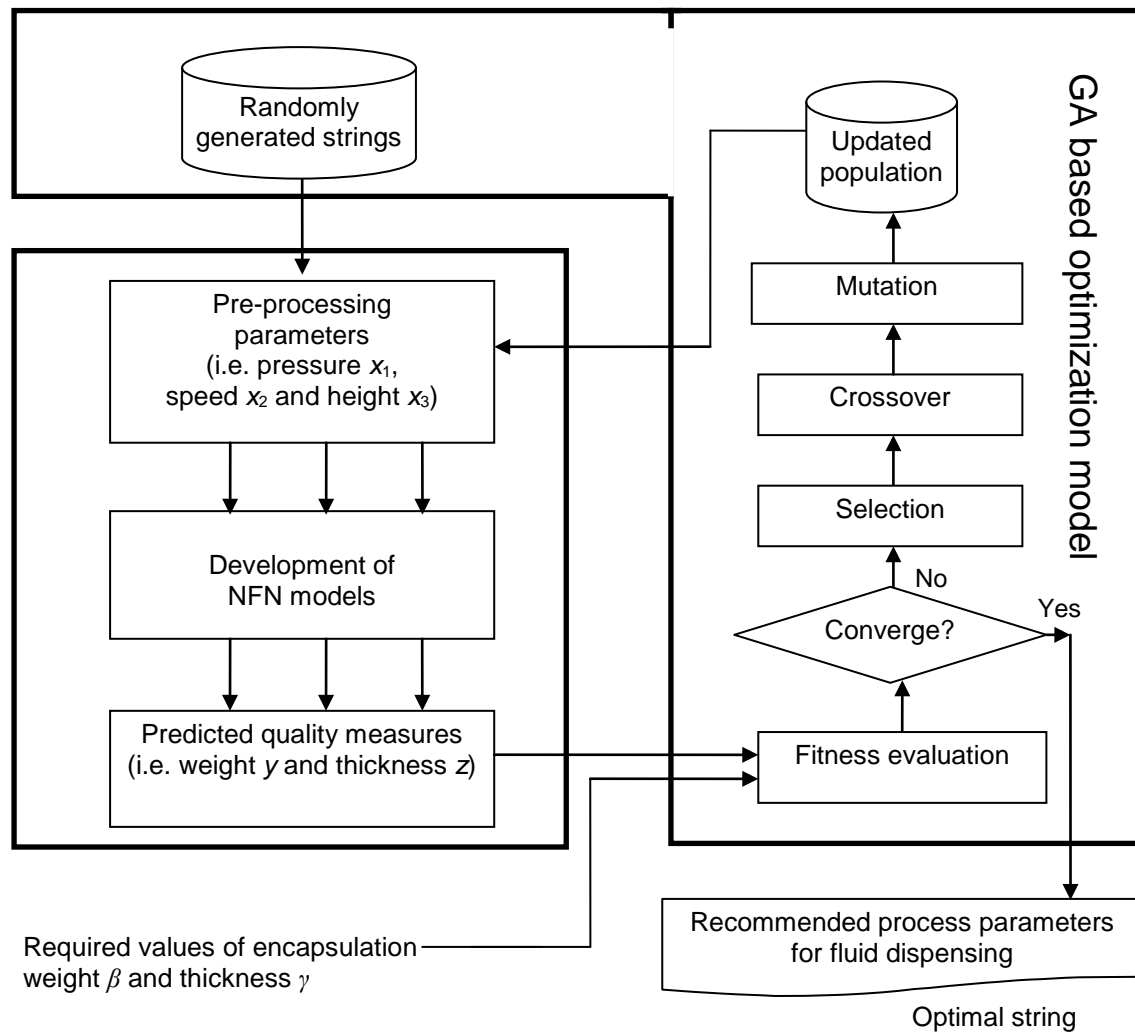


Figure 13 A GA based optimization model

(7) is a Pareto-based multi-objective problem which can be solved by GA which has rich literature in solving multi-objective problems (Knowles and Corne 2000, Zitzler et al 1999). Therefore GA was used in this research. Figure 13 shows a GA based optimization model for determining process parameters setting for the fluid dispensing and how it interfaces with the NFN based process models (as shown in Figure 2) was developed and implemented. The GA optimization model aims at minimizing the objective function (7) from which an optimal process parameters (x_1 , x_2 and x_3) setting can be obtained to achieve the required encapsulation weight

β mg and encapsulation thickness γ mm. Some descriptions of the genetic algorithm based optimization model are given below.

4.1 Randomly generated strings and fitness evaluation

The first step of genetic algorithm is to randomly generate strings which represent the population of genetic algorithm. Real and binary encoding are two commonly used approaches for string representation. In binary encoding representation, strings need to be encoded to real values for fitness evaluation and also they need to be decoded again for reproduction operation. However, in real encoding representation, there is no need for string encoding and decoding, which would save a considerable amount of computational time and would lead to a much faster searching process than that of the binary code representation. In this research, real encoding was adopted in this project, and each string contains three real values, which correspond to the setting of the three process parameters. The generated random values have the ranges associated with the operating ranges of the process parameters.

The multi-objective function (7) was formulated to determine how close the randomly generated strings are to the required values of encapsulation weight (β mg) and encapsulation thickness (γ mm). Based on a string with the three process parameters, the quality measures of encapsulation weight y and encapsulation thickness z can be predicted by the NFN based process models. Based on the fitness function (7), the fitness of a string with three process parameters can be evaluated.

4.2: Convergence and selection

The population is evolved and improved iteratively until a stopping condition is met. In genetic algorithms, there are several stopping conditions. In this research, the stopping criterion is met when the number of generations is equal to a pre-defined number of generations. Otherwise, the genetic algorithm goes to the next evolutionary generation.

The approach of roulette-wheel, which is one of the most common selection methods used for selecting strings to perform reproduction operations (Goldberg 1989), is used for strings selection. For example, the fitness of the j^{th} string in a population is assigned a value fitness_j . The fitness values are used to assign a probability value prob_j to the j^{th} string. The probability value prob_j is defined as:

$$\text{prob}_j = \frac{\text{fitness}_j}{\sum_{j=1}^{\text{Popsize}} \text{fitness}_j} \quad (8)$$

where Popsize is the population size of the genetic algorithm. Equation (8) shows that the string with a larger fitness value has higher probability to be selected.

4.3 Crossover, mutation and updating population

Intermediate crossover (Muhlenbein & Voosen 1993), a common crossover operation for real encoding representation, is used in the genetic algorithm. It is a method of producing a new string around and between the process parameters of the two selected parent strings. A new string $[x_1^3, x_2^3, x_3^3]$ is produced according to the following rule:

$$[x_1^3, x_2^3, x_3^3] = [x_1^2, x_2^2, x_3^2] \times \alpha \left\{ [x_1^2, x_2^2, x_3^2] - [x_1^1, x_2^1, x_3^1] \right\} \quad (9)$$

where α is a scaling factor chosen uniformly at random over some interval typically $[-0.25, 1.25]$, and $[x_1^1, x_2^1, x_3^1]$ and $[x_1^2, x_2^2, x_3^2]$ are the two selected parent strings. Process parameters in the new string are the result of combining the values of the process parameters in the parent strings according to (9) with a scaling factor α chosen for each process parameter. In geometric terms, intermediate crossover is capable of producing new parameter values within a slightly larger hypercube than that defined by the parent strings but constrained by the range of the scaling factor α .

Mutation is carried out by randomly changing one or more process parameter values of the selected string between its pre-defined upper and lower bound. The mutation operator of Gaussian perturbation of individual parameters is used. For example, the parameter x_j is selected to be mutated. After performing mutation, its value becomes:

$$x_j' = x_j + \text{MutMx} \times R_j \times \delta \quad (10)$$

where $\text{MutMx} = +1$ or -1 with equal probability; $R_j = 0.5 \times$ searching domain of the process parameter x_j ; $\delta =$ a value in the range $[0,1]$ for shrinking the mutation range based on Gaussian perturbation.

Updated population is produced by reinserting the new reproduced strings into the old population. It can be produced by replacing the least fitness strings in the old population with the new reproduced strings. However, pre-mature convergence likely occurs in early generations of evolution. To avoid the problem, random reinserting approach is used in the genetic algorithm.

5. Implementation and results

There are several evolutionary algorithm parameters such as population sizes, crossover rates and mutation rates that need to be set such that evolutionary algorithms can find solutions efficiently [Back and Schwefel 1993, Eiben et al 1999]. It can be found that different results could be obtained using different settings of genetic algorithm parameters. Although there are a lot of studies of investigating performance of genetic algorithms using different settings of GA parameters [Back and Schwefel 1993, Eiben et al 1999, Goldberg 1989, Grefenstette 1986, De Jong and Spears 1990], no conclusion has been reached by which a parameter setting outperform the others on which problem landscapes. As this research project focuses on modeling and optimization of fluid dispensing using neural fuzzy networks and GAs, we adopted DeJong's suggestions of algorithm parameters settings [DeJong 1975] in this research. Referring to the

DeJong's suggestion, population size, crossover rate, the mutation rate and pre-defined number generations are set as 200, 0.8, 0.001 and 200 respectively. The genetic algorithm toolbox coded in Matlab (Chipperfield et al. 1994), which exists in the public domain, has been employed to search the process parameters of the fluid dispensing process.

Eight validation tests were performed to evaluate the effectiveness of the GA based optimization model. For each test, required values of encapsulation weight and thickness are input and it is expected that a setting of process parameters can be obtained from the optimization model. Table 5 shows the eight settings of process parameters generated by the GA based optimization model corresponding to the eight sets of requirements of quality characteristics. The second and third columns show the required values and the predicted values of encapsulation weight and encapsulation thickness. The last three columns show the settings of the process parameters generated by the GA based optimization model. From the table, it can be found mean prediction errors between the required values and the predicted values of the encapsulation weight and encapsulation thickness are all about 3%.

As an instance, the distributions of chromosomes the GA run of the 1st validation test at the 5-th, 50-th, 200-th and 500-th are shown in Fig. 14, 15, 16 and 17 in the Appendix. It can be found that the chromosomes distribute more randomly at the 5-th, 50-th and 200-th than the one at the 500-th. It can also be observed that the shape of a Pareto – optimum occurs at the 500-th generation in which both the error of encapsulation weight of chromosome and the error of encapsulation thickness of chromosome are more equally than the ones at the 5-th, 50-th and 200-th generations.

Table 5 Recommended process conditions with respect to the requirements of quality characteristics

	Requirements of quality characteristics		Validation results of process optimization						
	Weight β (mg)	thickness γ (mm)	Weight β (mg)		Thickness γ (mm)		Recommended process condition		
			Opt. value (mg)	Relative error (%)	Opt. value (mm)	Relative error (%)	x_3 (bar)	x_2 (steps)	x_3 (rpm)
1	72.3	0.58	74.6358	3.2308	0.5953	2.6327	2.9655	472.0325	633.9157
2	43.2	0.48	45.1413	4.4937	0.4851	1.0688	2.0460	903.9330	898.3743
3	87.4	0.67	84.4016	3.4307	0.6251	6.7015	4	2000	666.38
4	37.2	0.46	36.1849	2.7287	0.4471	2.8143	1.5522	1282.7	970.1482
5	75.1	0.62	71.1691	5.2342	0.5789	6.6237	1.2595	1536.4	497.8665
6	59.3	0.57	60.7858	2.5055	0.5537	2.8624	1.1937	357.7756	574.6837
7	62.4	0.53	63.4488	1.6808	0.51395	3.0283	1.6594	250	613.77
8	53.1	0.53	53.4702	0.6972	0.5292	0.1509	3.9027	830.9390	958.8334
mean relative errors (%)			3.0002		3.2353				

6. Conclusion

In this paper, neural fuzzy networks and genetic algorithms were proposed to determine process parameters setting of fluid dispensing process for microchip encapsulation. 96 experiments based on a full factorial design were conducted in which the parameters of compressed air pressure, pump motor speed, distance between substrate and needle, and two quality characteristics, encapsulation weights and encapsulation thickness, are involved. 88 experimental data were used to develop the NFN based process models. 8 validation tests were carried out to evaluate the effectiveness and robustness of the process parameters settings recommended by the GA based optimization model. The validation results indicate that, for each test, a setting of process parameters can be obtained to achieve the required quality requirement based on the proposed approaches.

Further work involves the validation of the process conditions obtained by the system in a real world environment using experiments. Improvement of the NFN based process models and

the GA based optimization model in terms of the computational speed and accuracy could be considered in future studies. A data mining rule-based system will be developed to reduce the searching space of the GA based optimization model. The results will be reported in the near future.

References

- Babets K., Geskin E.S., 2000. Application of fuzzy logic for modeling of waterjet depainting. *Machining Science and Technology* 4(1), 81-100.
- Back T., Schwefel H.P., An overview of evolutionary algorithms for parameter optimization, *Evolutionary Computation*, vol. 1, no. 1, pp. 1-23, 1993.
- Brinksmeier E., Tonshoff H.K., Czenkusch C., Heinzl C., 1998. Modelling and optimisation of grinding processes. *Journal of Intelligent Manufacturing* 9, 303-314.
- Chen D.X., 2002. *Modeling and off-line control of fluid dispensing for electronics packaging*. PhD thesis, The University of Saskatchewan.
- Chen Y., Tang J., Fung R.Y.K., Ren Z., 2004. Fuzzy regression based mathematical programming for QFD. *International Journal of Production Research* 42(5), 3583-3604.
- Chiang H. H., Hieber C. A., Wang K.K., 1991. A unified simulation of the filling and postfilling stages in injection molding. Part 1: Formulation. *Polymer Engineering and Science* 31, 116-124.
- Chipperfield A. J., Fleming P. J., Pohlheim H., Fonseca C. M., 1994. Genetic Algorithm Toolbox User's Guide. *ACSE Research Report No. 512*, University of Sheffield.
- DeJong K.A., 1975. *An analysis of the behaviour of a class of genetic adaptive systems*. PhD thesis, Department of Computer and Communication Science, University of Michigan.

- De Jong K. A., Spears W. M., 1990. An analysis of the interacting roles of population size and crossover in genetic algorithms, *The First International Conference on Parallel Problems Solving from Nature*, 38-47.
- Eiben A.E., Hinterding R., Michalewicz Z., 1999. Parameter control in evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124-141.
- Fiordaliso A., 2001. A constrained Takagi-Sugeno fuzzy system that allows for better interpretation and analysis. *Fuzzy Sets and Systems* 118, 307-318.
- Goldberg D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. United States of America: Addison Wesley Longman, Inc.
- Grefenstette J.J., 1986. Optimization of control parameters for genetic algorithms, *IEEE Transaction on Systems, Man and Cybernetics* 16(1), 122-128.
- Han R., Shi L. and Gupta M., 2000. Three-dimensional simulation of microchip encapsulation process. *Polymer Engineering and Science* 40(3), 776-785.
- Hussein Y.A. and El-Ghazaly S.M., 2004. Modeling and optimization of microwave devices and circuits using genetic algorithms. *IEEE Transactions on Microwave Theory and Techniques* 52(1), 329-336.
- Ip, K. W., Kwong, C. K., Bai, H., and Tsim, Y. C., 2003. Process modelling of epoxy dispensing for microchip encapsulation using fuzzy linear regression with fuzzy intervals. *International Journal of Advanced Manufacturing Technology* 22, 417-423.
- Kang, S. Y., Xie, H., and Lee, Y. C., 1993. Physical and fuzzy logic modeling of a flip-chip thermocompression bonding process. *Journal of Electronic Packaging* 115, 63-70.
- Knowles J. D., Corne D.W., 2000. Approximating the nondominated front using the pareto archived evolution strategy, *Evolutionary Computation* 8, 149-172.

- Kwong C.K., Chan K.Y., Wong H., 2007. An empirical approach to modeling fluid dispensing for electronic packaging. *International Journal of Advanced Manufacturing Technology* 34, 111-121.
- Lai Y.J., Chang S.I., 1994. A Fuzzy Approach for Multiresponse Optimization: An off-line Quality Engineering Problem. *Fuzzy Sets and Systems* 63, 117-129.
- Li H.L., Chou T., C.P. Chou., 2007. Optimization of resistance spot welding process using Taguchi method and a neural network. *Experimental Techniques* 31(5), 30-36.
- Li H.X., Tso S.K., Deng H., 2001. A concept approach to integrate design and control for the epoxy dispensing process. *International Journal of Advanced Manufacturing Technology* 17, 677-682.
- Ling C.H., Leung F.H.F., Lam H.K., Tam P.K.S., 2003. Short-term electric load forecasting based on a neural fuzzy network. *IEEE Transactions on Industrial Electronics* 50(6), 1305-1316.
- Muhlenbein H., 1992. How genetic algorithms really work-Part I: Mutation and hillclimbing, *Proceedings of the 2nd International Conference on Parallel Problem Solving from Nature*, 15-25.
- Muhlenbein H., Voosen D.S., 1993. Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation* 1(1), 25-49.
- Petri K.L., Billo R.E., Bidanda B., 1998. A neural network process model for abrasive flow machining operations. *Journal of Manufacturing Systems* 17(1), 52-65.
- Poon G.K.K., Williams D.J., Chin K.S., 2000. Optimizing the lithographic patterning effect in an acid copper electroplating process. *International Journal of Advanced Manufacturing Technology* 16, 881-888.

- Schaffer J., Caruana R., Eshelman L., Das R., 1989. A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proceedings of the 3rd International Conference on Genetic Algorithms*, 51-61.
- Schaible B., Lee Y.C., 1997. Fuzzy Logic Based Regression Models for electronics manufacturing applications. *Advances in Electronic Packaging* 1, 147-155.
- Seber G.A.F., 2003. *Linear regression analysis*, Wiley.
- Tan K. H., Yuen M.M.F., 2000. A fuzzy multiobjective approach for minimization of injection molding defects. *Polymer Engineering and Science* 40(4), 956-971.
- Tong K.W., Kwong C.K. and Yu K.M., 2004. Intelligent process design system for the transfer moulding of electronic packages. *International Journal of Production Research* 42(10), 1911-1931.
- Tzafestas S. G., Zikidis K. C., 2001. NeuroFAST: on-line neuro-fuzzy ART-based structure and parameter learning TSK model. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics* 31(5), 797–802.
- Wilson P.R., Ross J.N., Brown A.D., 2001. Optimizing the Jiles-Atherton model of hysteresis by a genetic algorithm. *IEEE Transactions on Magnetics* 37(2), 989-993.
- Xie H., Lee Y. C., 1994. Process optimization using a fuzzy logic response surface method. *IEEE Transactions on Components, Packaging and Manufacturing Technology – Part A* 17(2), 202-210.
- Xie H., MAHAJAN R. L., LEE Y.C., 1995. Fuzzy logic models for thermally based microelectronic manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing* 8(3), 219-226.
- Yarlagadda P.K.D.V., 2000. Prediction of die casting process parameters by using an artificial neural network model for zinc alloys. *International Journal of Production Research* 38(1), 119-139.

Ying H., 1998. General MISO Takagi-Sugeno fuzzy systems with simplified linear rule consequent as universal approximators for control and modelling applications. *Information Sciences* 108, 91-107.

Zitzler E., Thiele L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, *IEEE transactions on Evolutionary Computation* 3(4), 257-271.

Appendix

Table 1 N-fold validation on encapsulation weight (mean errors)

Trials	1	2	3	4	5	6	7	8	9	10	11	mean
SR	2.6167	5.5694	4.4299	6.9069	5.9278	5.2842	7.8274	5.3838	3.3115	7.0466	3.5837	5.2625
FR	9.5184	5.9854	8.2059	8.7820	21.3973	8.8987	14.4420	13.6757	5.7829	15.1002	8.9840	10.9793
NN	2.8521	4.7557	4.1000	6.6230	6.4458	4.8087	7.5392	5.1239	3.0325	6.7104	3.4329	5.0386
NFN	2.5411	4.7666	3.9463	5.9725	5.7491	4.6673	6.9703	4.9626	2.8342	6.3694	3.0980	4.7161

Table 2 N-fold validation on encapsulation weight (variances of errors)

Trials	1	2	3	4	5	6	7	8	9	10	11	mean
SR	2.0329	4.1817	4.6365	6.2247	3.7489	5.1607	10.1716	3.5386	5.2920	5.9655	3.0582	4.9101
FR	8.3080	4.1697	5.3148	6.1757	21.0245	7.5905	9.0266	14.9656	4.8712	11.7089	10.7331	9.4444
NN	1.9908	2.9812	3.3651	4.4350	4.1630	3.8899	7.1553	3.5088	3.7378	4.7496	2.8427	3.8927
NFN	1.9543	2.9659	3.3449	4.4124	4.0692	3.8598	7.1234	3.4429	3.7204	4.7011	2.7959	3.8537

Table 3 N-fold validation on encapsulation thickness (mean errors)

Trials	1	2	3	4	5	6	7	8	9	10	11	mean
SR	5.0367	5.1956	4.2001	5.2102	2.3794	4.3085	4.3628	4.1777	3.7339	4.7027	2.7146	4.1838
FR	8.7428	10.3850	8.4335	7.0995	9.7924	9.6185	10.9179	7.6667	9.7215	8.8324	7.4376	8.9680
NN	4.7145	4.9939	4.0407	4.6909	2.7688	4.2376	4.4037	3.9501	3.8050	4.4659	2.8017	4.0793
NFN	4.3260	4.6079	3.7291	4.2680	2.6463	3.9283	4.1032	3.6324	3.5519	4.1104	2.6216	3.7750

Table 4 N-fold validation on encapsulation thickness (variances of errors)

Trials	1	2	3	4	5	6	7	8	9	10	11	mean
SR	5.3338	5.1579	3.1694	6.1514	1.7856	2.7014	2.9673	4.6335	3.2039	4.8416	1.7221	3.7880
FR	5.4407	3.0829	4.9998	6.1072	3.5297	5.6476	6.0247	4.4292	4.7949	2.9651	4.9830	4.7277
NN	4.3779	4.0619	2.7407	5.0358	1.6002	2.4452	2.6718	3.7797	2.7500	3.8184	1.6677	3.1772
NFN	4.2842	3.9648	2.6901	4.9272	1.5738	2.4062	2.6284	3.6973	2.6982	3.7275	1.6469	3.1131

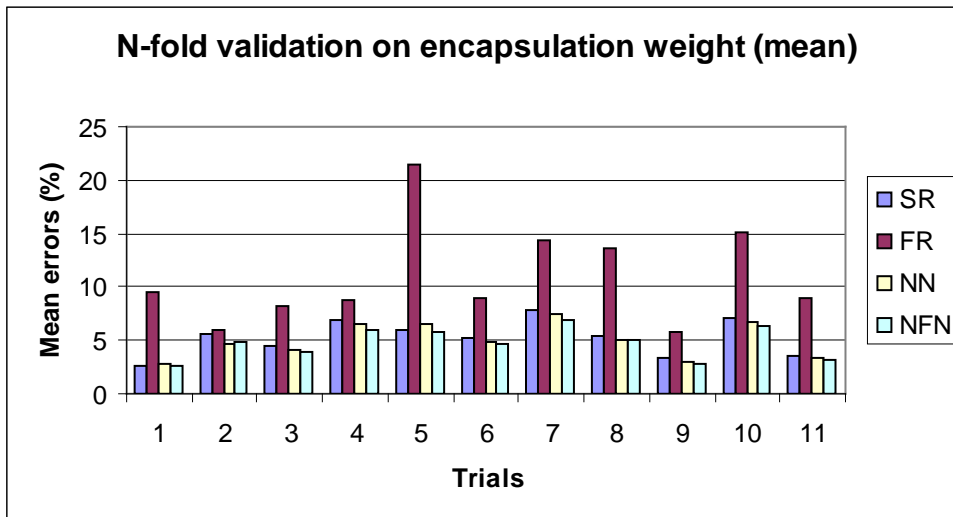


Figure 9 N-fold validation on encapsulation weight (mean errors)

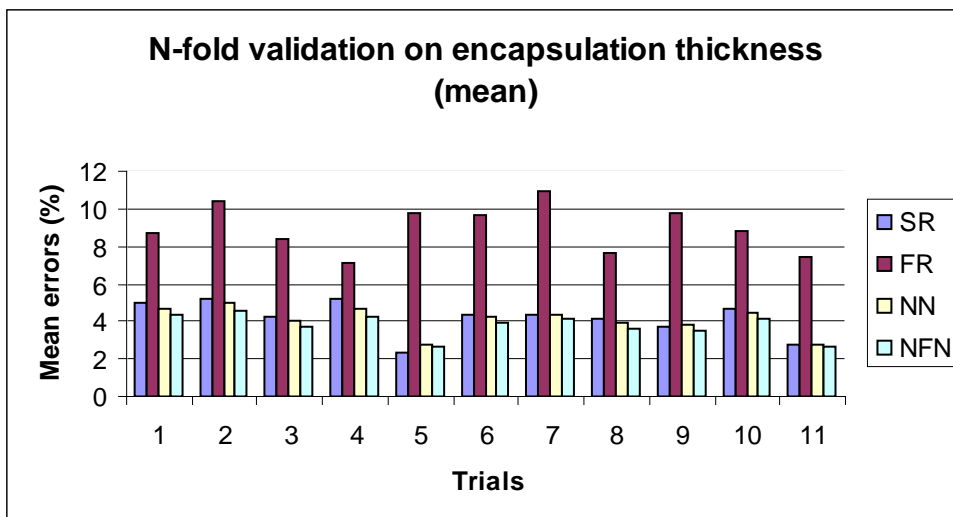


Figure 10 N-fold validation on encapsulation weight (variances of errors)

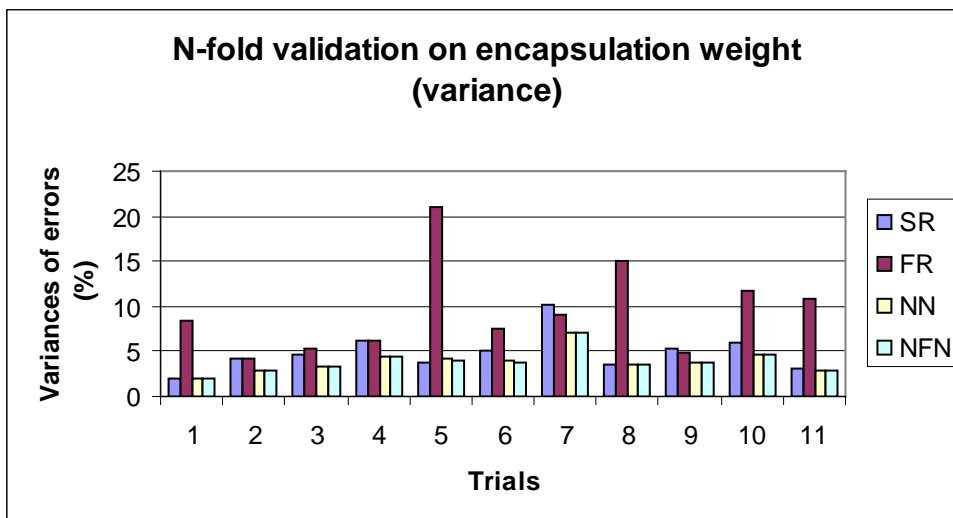


Figure 11 N-fold validation on encapsulation thickness (mean errors)

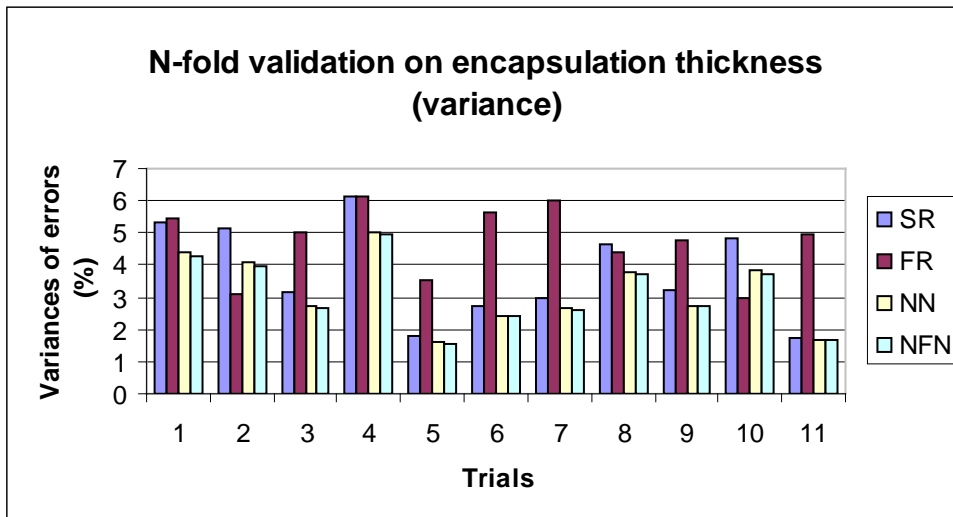


Figure 12 N-fold validation on encapsulation thickness (variances of errors)

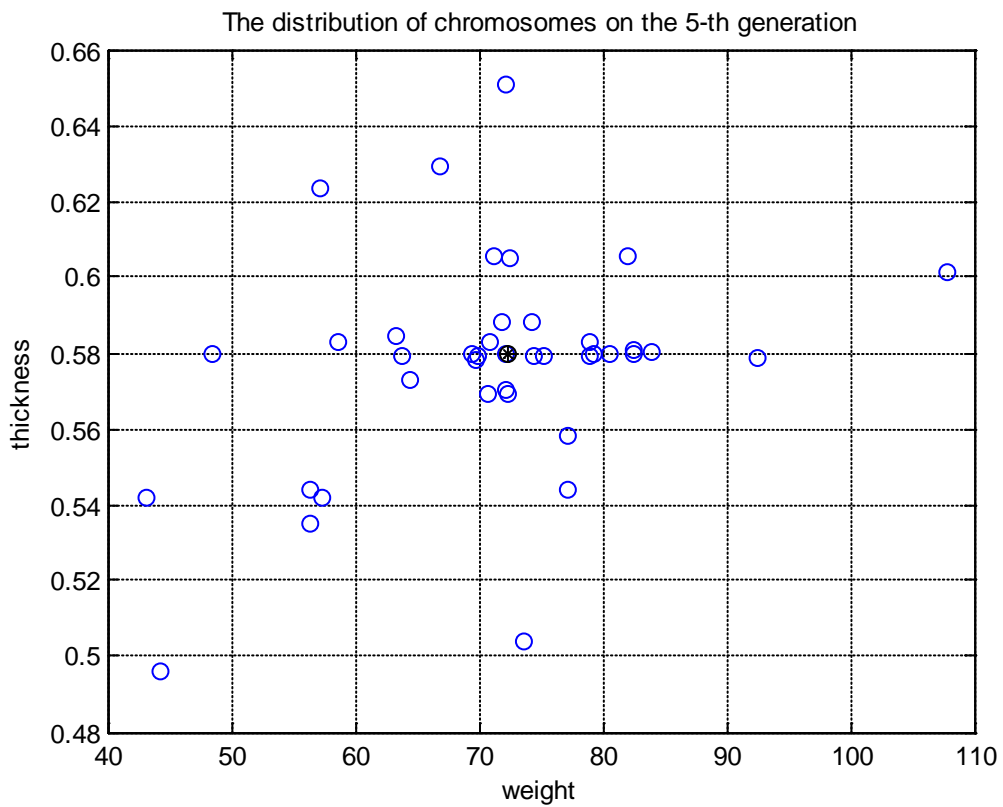


Figure 14 Distribution of chromosomes at the 5-th generation

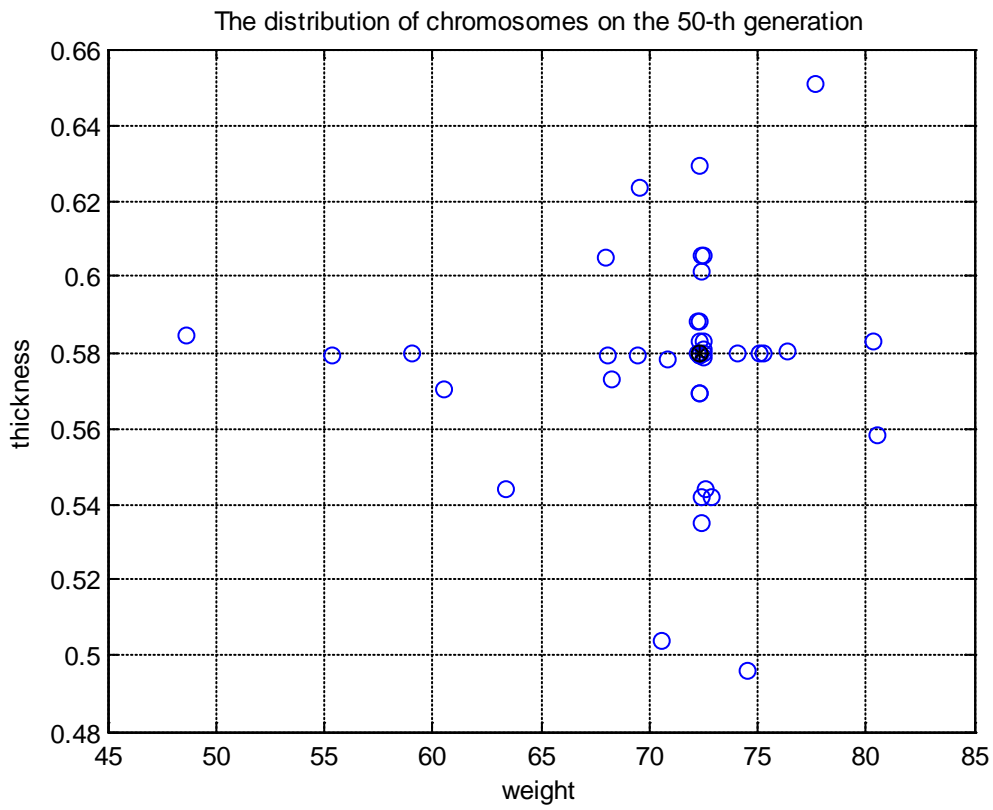


Figure 15 Distribution of chromosomes at the 50-th generation

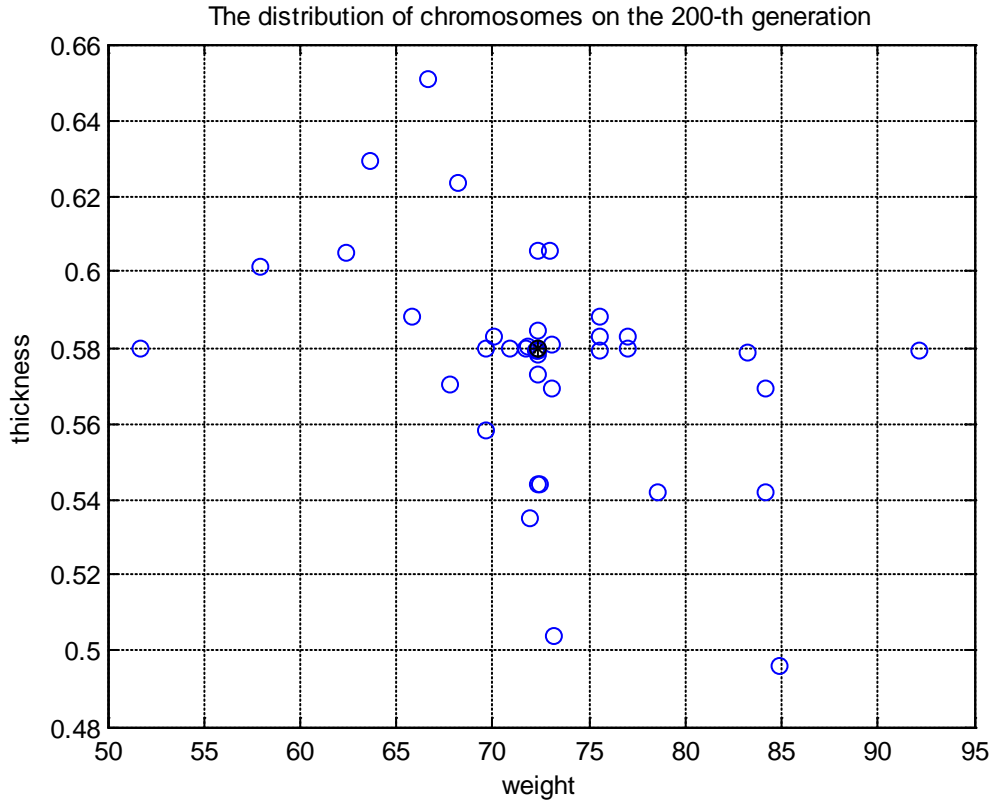


Figure 16 Distribution of chromosomes at the 200-th generation

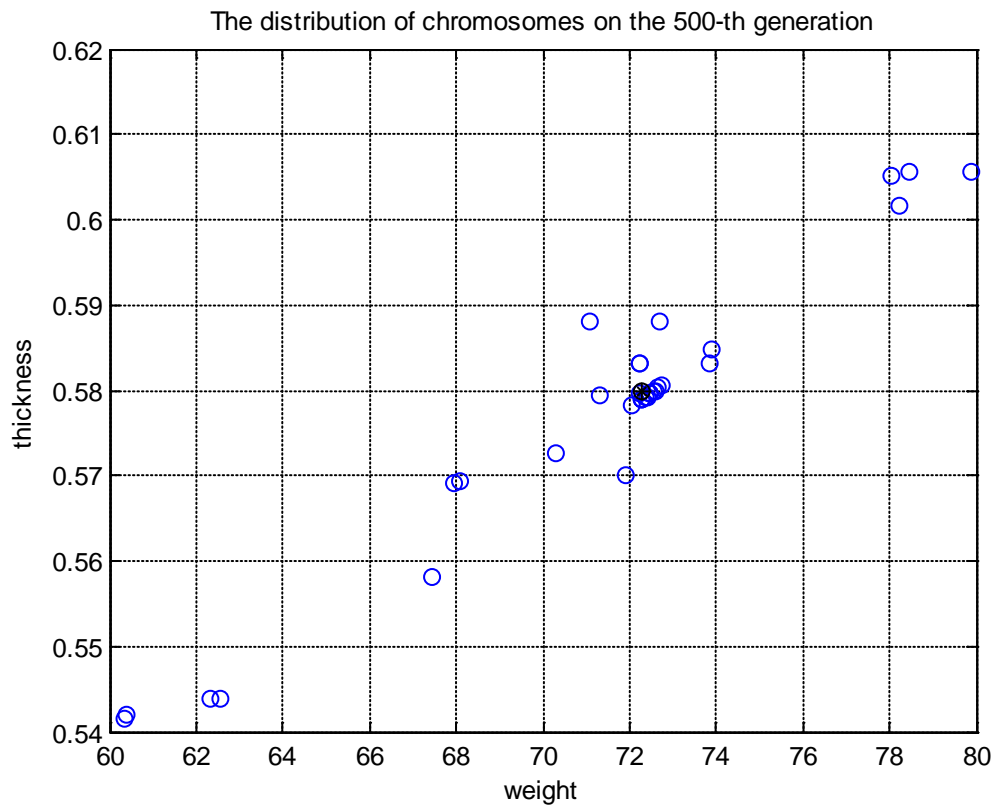


Figure 17 Distribution of chromosomes at the 500-th generation