International Journal of Applied Science and Engineering
2004. 2, 1: 59-71

Process Optimization by Soft Computing and Its Application to a Wire Bonding Problem

Chi-Bin Cheng

Department of Industrial Engineering and Management,
Chaoyang University of Technology,
Wufeng, Taichung country 413, Taiwan, R.O.C.

Abstract: Modeling and optimization of a process with multiple outputs is discussed in this paper. A neuro-fuzzy system named MANFIS, which comprises a fuzzy inference structure and neural network learning ability, is used to model a multiple output process. Optimization of such a process is formulated as a multiple objective decision making problem, and a genetic algorithm and a numerical method are introduced, respectively, to solve this problem based on the MANFIS model. We have used these two algorithms, respectively, to solve a chemical process optimization problem, and compared their performances. A combination of these two algorithms is also suggested to improve performances of both algorithms. The proposed approach is also applied to a wire-bonding problem in semiconductor manufacturing.

Keywords: process optimization; soft computing; neuro-fuzzy system; genetic algorithm, multiple objective decision making; wire bonding.

1. Introduction

Process optimization is to find a setting of the controllable variables, or called input variables, so as to obtain the best outcomes of a process. The outcomes of a process are often referred to as the outputs or responses of a system. In this study, optimization of a process with multiple outputs is considered. For example, in a tool life problem, we attempt to determine the cutting speed and depth of cut so as to obtain a maximal life of the tool (a primary response) and retain a satisfied rate of metal removed (a secondary response) at the same time.

Response optimization methods are popular tools for process optimization. Usually, these methods include two stages. In the first stage, we use regression analysis to model a system's responses; that is, we identify the relationship between responses and input variables through regression functions. In the second stage, we use optimization techniques to obtain a setting of system parameters that give system the most desirable responses. Traditionally, in the first stage, linear regression with the regressors in a first-order or second-order polynomial form is used to approximate the response surfaces. However, frequently encountered in practice, systems are complicated and highly nonlinear, and thus, linear regression is not suitable. In recent years, nonparametric regression approaches, such as neural networks and fuzzy inference systems, are widely adopted for modeling nonlinear systems. These nonparametric approaches learn the relations between input variables and responses directly from

Corresponding e-mail: cbcheng@mail.cyut.edu.tw
Accepted for Publication: Dec. 23, 2003

the observations without assuming any pre-specified functional form. The combination of fuzzy inference systems and neural networks, together with genetic algorithms, create a new research area called soft computing. Soft computing emerges as a computing approach that tries to mimic human's ability of reasoning and learning in an uncertain environment. One representative technique of soft computing is neuro-fuzzy systems. A neuro-fuzzy system is a fuzzy inference system presented in a network structure, and equipped with neural network learning abilities. In this study, we use a neuro-fuzzy system, named multiple adaptive neuro-fuzzy inference system (MANFIS) [7], to model a system that has multiple responses; and furthermore, we also use a genetic algorithm (GA) [5] to optimize this system's responses based on the model of MANFIS.

A multiple response system has m response $y_1, y_2,..., y_m$, which are affected by a set of input variables $\mathbf{x} = (x_1, x_2,..., x_p)^T$. Traditionally, the relations between responses and input variables are defined through functions:

$$y_i = f_i(\mathbf{x}) + \varepsilon_i,\ i = 1, 2, ..., m, \qquad (1)$$

where f_i is the functional relation between $\mathbf{x}$ and the i-th response y_i, and ε_i are i.i.d. random errors with zero means and constant variances σ_i^2, $\forall i$. The objective of the multiple response optimization is to find a solution $\mathbf{x}^*$ such that each response will attain a compromised optimum.

Many approaches have been proposed to solve the multiple response optimization problem. Derringer and Suich [4] transform each response function into a desirability function, and then maximize the geometric mean of the individual desirability functions to obtain a compromised solution. Khuri and Conlon [8] presented a procedure based on a distance function that calculates the overall closeness where the response functions achieve their respective optimum at the same set of conditions; a compromised solution is then found by minimizing this distance function over the experimental region. Pignatiello [13], Ames et al. [1] and Vining [16] all propose to minimize a measure based on a multivariate loss function, which evaluates the loss when responses deviate from their targets. For the special case of two responses, Myers and Carter [11] introduced a dual response approach, which optimizes the primary response subject to an appropriate constraint on the secondary response. The disadvantage of their approach is that such an optimization scheme can be misleading due to the unrealistic restriction of forcing the constrained response to a specific value [10]. To remedy the disadvantage of the approach of Myers and Carter [11], Kim and Lin [9] formulate the dual response problem as a multiple objective decision making (MODM) programming and introduce a fuzzy optimization methodology, which is based on Zimmermann's maximin approach [17]. Their approach optimizes the primary response and the secondary response, simultaneously, by maximizing a compromised satisfaction degree of both responses. The degrees of satisfaction of both the mean response and deviation are defined by membership functions originated in fuzzy set theory. Though the previous approaches varied in their solution procedures, they commonly assumed linear response surfaces.

In this study, to deal with nonlinear responses, the system is modeled by MANFIS, and the multiple response optimization problem is formulated as an MODM. However, since we use the nonparametric regression tool MANFIS to model responses, exact functional forms of responses are not known and hence derivative-based optimization methods cannot be directly applied to obtain the optimal solution. Therefore, we will use a genetic algorithm as well as a numerical method to search optimal solutions on the response surfaces. Performances of these two algorithms will be compared.

The remainder of this paper is organized as follows. In the next section, the architecture

of MANFIS and its learning process are summarized. Section 3 formulates the multiple response optimization problem as an MODM. Two optimization algorithms, a genetic algorithm and a numerical method, are presented in Section 4 to solve the MODM. For illustration, Section 5 uses the two algorithms, respectively, to solve a chemical process optimization problem. To improve the performances of both algorithms, we also suggest a combination of these two algorithms. Computational results show this combined algorithm is promising. Concluding remarks are given in the last section.

2. Multiple adaptive neuro-fuzzy inference system

MANFIS is an extension of the sin gle-output neuro-fuzzy system ANFIS [6], for producing multiple outputs. A neuro-fuzzy system can serve as a nonparametric regression tool, which model the regression relationship non-parametrically without reference to any pre-specified functional form. MANFIS can be viewed as an aggregation of many independent ANFIS. The architecture of MANFIS is depicted in Figure 1.

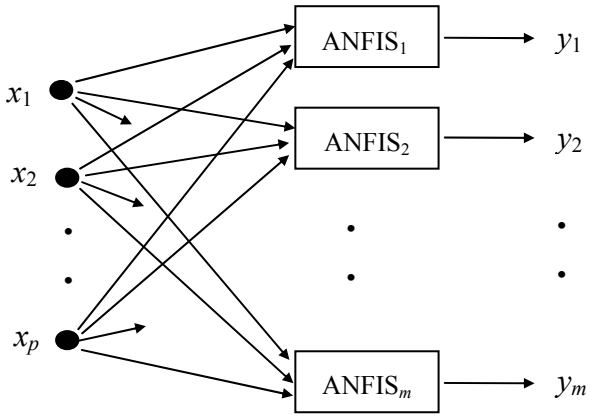

Figure 1. Architecture of MANFI S

Every single ANFIS in an MANFIS simulates the functional relations f_i, i=1,..., m, in Equation (1). ANFIS can be considered as a network presentation of a TSK fuzzy inference system [15], and the if-then rules in TSK are comprised in the network structure. To illustrate the architecture of ANFIS, an example with a two-dimensional input is visualized in Figure 2.

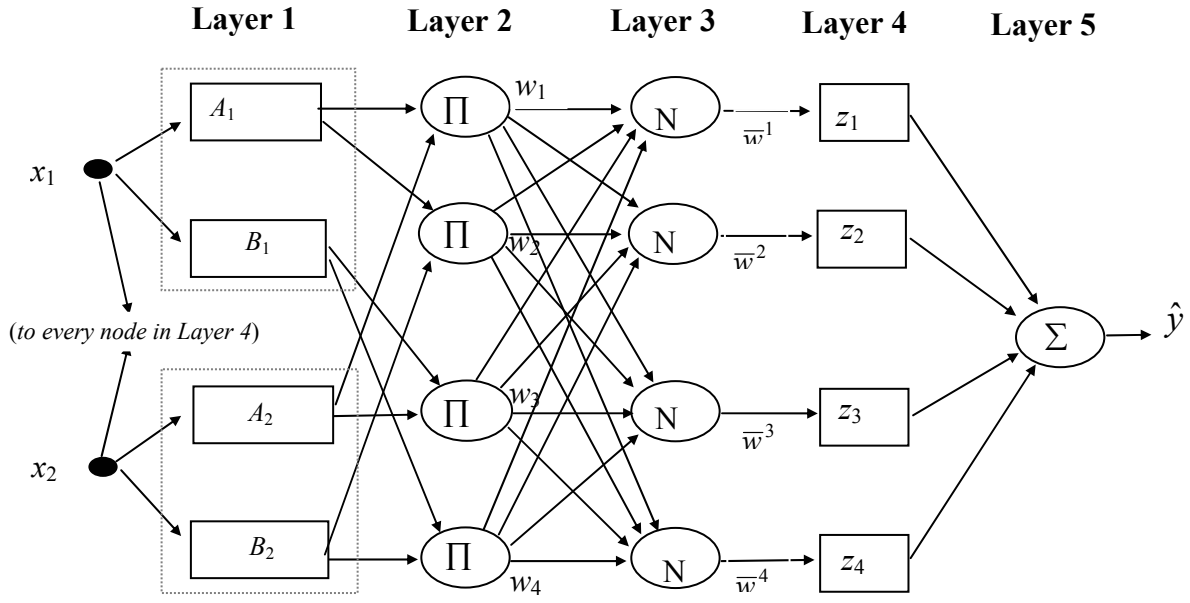

Figure 2. Architecture of ANFIS

To reflect different adaptive capabilities, the nodes in ANFIS are represented by circles or squares, in which, square nodes represent adaptive nodes and circle nodes represent fixed nodes. Adaptive nodes contain parameters that can be adjusted by learning, while the fixed nodes do not contain adjustable parameters. In this study, the adaptive nodes in layer 1 of the ANFIS are parameterized by Gaussian functions with their means and deviations. Nodes in layer 2 are fixed nodes labeled Π, which is a fuzzy conjunction operator. Functions of nodes in this layer are to synthesize the information from the first layer. The operator Π is defined as a multiplication of all of its incoming signals, and output the firing strength w_j, j=1,...,4. Nodes in layer 3 labeled by N simply performs a normalization of signals from layer 2 and output the normalized firing strength $\overline{w}_j = w_j / \sum_{r=1}^{4} w_r$, j=1,...,4. The adaptive nodes in layer 4 of the ANFIS contain linear functions of the input variables with their coefficients as the adjustable parameters; that is, $z_j = a_j x_1 + b_j x_2 + c_j$, j=1,...,4. The single node in layer 5 is a fixed node, which computes the overall output as the summation of all incoming signals: $\hat{y} = \sum_{j=1}^{4} \overline{w}_j z_j$.

Assuming that we have conducted an experiment with n runs on an m-response system, n observations are collected with the format of $(\mathbf{x}_k, y_{1k},...,y_{ik},...,y_{mk})$, k=1,...,n, where $\mathbf{x}_k$ is the input condition at the k-th run and y_{ik} is the i-th response at the k-th run. With these observations, MANFIS can approximate the multiple responses y_i, i=1,...,m, by minimizing an error measure E defined as

$$E = \sum_{k=1}^{n} \sum_{i=1}^{m} (y_{ik} - \hat{y}_{ik})^2, \tag{2}$$

where $\hat{y}_{ik}$ is the estimate of the i-th response for the k-th run. The minimization of E is carried out in an iterative manner, which is referred to as a learning process. The learning process of MANFIS terminates when the error measure E reduces to a satisfactory level. Since E is a summation of the squared errors from m independent ANFIS, the learning of MANFIS can be treated as the learning of m independent ANFIS. Furthermore, since ANFIS is a multi-layered-feed-forward network, backpropagation learning algorithms used in neural networks can be directly applied to its learning. The details of this learning process can be found in [6].

3. MODM formulation of multiple response optimization

By means of the learning process, MANFIS obtains an estimation of desired outputs with given inputs. Let $\hat{y}_i$, i=1,...,m, be the i-th output of MANFIS, and they are estimates of multiple responses $y_1,...,y_m$, respectively. To indicate these estimates are functions of the input variables $\mathbf{x}$, they will be denoted as $\hat{y}_i(\mathbf{x})$, i=1,...,m.

Since the system under discussion has multiple responses, the optimization of the system in fact involves the optimization of several individual responses at the same time. For all the system responses, they can be divided into three sets: 1) the-larger-the-better, denoted by L; 2) the-smaller-the-better, denoted by S; and 3) the-nominal-the-best, denoted by N. We have formulated this optimization problem as a multiple objective decision making problem with the following form [2]:

$$\begin{aligned} &\max \hat{y}_l(\mathbf{x}), \forall l \in L \\ &\min \hat{y}_s(\mathbf{x}), \forall s \in S \\ &\min \left| \hat{y}_t(\mathbf{x}) - T_t \right|, \forall t \in N \\ &\text{s.t. } \mathbf{x} \in B, \end{aligned} \tag{3}$$

where T_t is the nominal target of the t-th response; and B is a feasible region of $\mathbf{x}$.

To solve the above multiple objective op-

timization problem, we follow the idea of Zimmermann's maximin approach [17]. According to the maximin approach, the solution of (3) can be obtained by maximizing an overall satisfactory degree among all individual objectives in (3). That is, for each objective, it has its own satisfactory degree, and the overall satisfaction is an intersection of all individual satisfactory degrees, where the intersection is defined through a min operator. The satisfactory degree for each objective is evaluated by an user-defined membership function $\mu_{\hat{y}_i}(\hat{y}_i)$. Let λ be the overall satisfactory degree, and then we can convert the original MODM (3) to:

$$\begin{aligned} &\max \lambda \\ &\text{s.t. } \mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x})) \geq \lambda, i = 1, ..., m, \\ &\mathbf{x} \in B, \\ &\lambda \in [0,1]. \end{aligned} \tag{4}$$

Each response's membership function $\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x}))$ should be well chosen so as to reflect its characteristic. For the response belonged to the set of the-larger-the-better, its degree of satisfaction reaches 1 when it is at $\hat{y}_i^* = \max_{\mathbf{x}\in B}\{\hat{y}_i(\mathbf{x})\}$ and then decreases monotonically to 0 at $\hat{y}_i^- = \min_{\mathbf{x}\in B}\{\hat{y}_i(\mathbf{x})\}$. A typical membership function for $\hat{y}_i(\mathbf{x})$, $i \in L$, could be stated as

$$\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x})) = \begin{cases} 1, & \text{if } \hat{y}_i(\mathbf{x}) > \hat{y}_i^*, \\ \dfrac{\hat{y}_i(\mathbf{x}) - \hat{y}_i^-}{\hat{y}_i^* - \hat{y}_i^-}, & \text{if } \hat{y}_i^- \leq \hat{y}_i(\mathbf{x}) \leq \hat{y}_i^*, \forall i \in L. \\ 0, & \text{if } \hat{y}_i(\mathbf{x}) < \hat{y}_i^-, \end{cases} \tag{5}$$

The above membership function is graphically shown in Figure 3.

For the response belonged to the set of the-smaller-the-better, we set the satisfactory degree to 1 when a response is at $\hat{y}_i^-$ and then it decreases monotonically to 0 at $\hat{y}_i^*$. Such type of membership functions can be expressed as

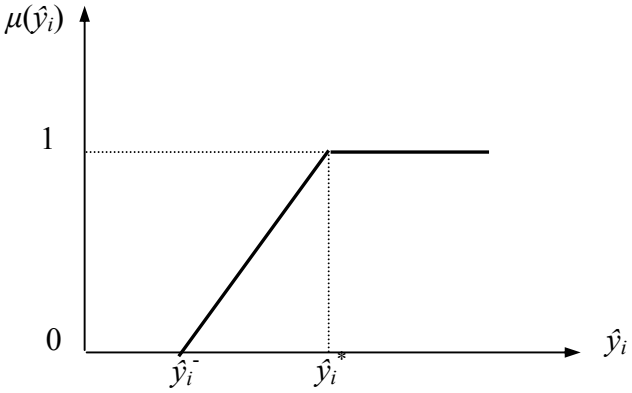

Figure 3. Membership function of $\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x}))$: "the larger the better" case

$$\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x})) = \begin{cases} 1, & \text{if } \hat{y}_i(\mathbf{x}) < \hat{y}_i^*, \\ \dfrac{\hat{y}_i^* - \hat{y}_i(\mathbf{x})}{\hat{y}_i^* - \hat{y}_i^-}, & \text{if } \hat{y}_i^- \leq \hat{y}_i(\mathbf{x}) \leq \hat{y}_i^*, \forall i \in S, \\ 0, & \text{if } \hat{y}_i(\mathbf{x}) > \hat{y}_i^*. \end{cases} \tag{6}$$

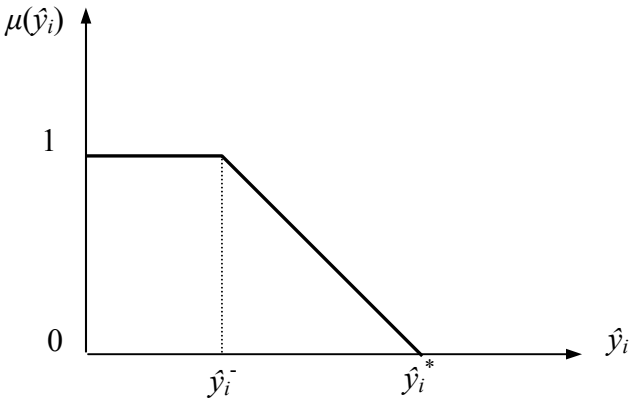

Figure 4. Membership function of $\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x}))$: "the smaller the better" case

Similarly, for the response of the set of the-nominal-the-best, the degree of satisfaction is maximized when it is at its target T_i, and decreases as it is away from T_i. Membership functions of this type can be defined as

$$\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x})) = \begin{cases} 1 - \dfrac{T_i - \hat{y}_i(\mathbf{x})}{T_i - \hat{y}_i^-}, & \text{if } \hat{y}_i^- < \hat{y}_i(\mathbf{x}) \le T_i, \\ 1 - \dfrac{\hat{y}_i(\mathbf{x}) - T_i}{\hat{y}_i^* - T_i}, & \text{if } T_i \le \hat{y}_i(\mathbf{x}) < \hat{y}_i^*, \forall i \in N, \\ 0, & \text{elsewhere} \end{cases} \quad (7)$$

Membership function of (7) is depicted in Figure 5.

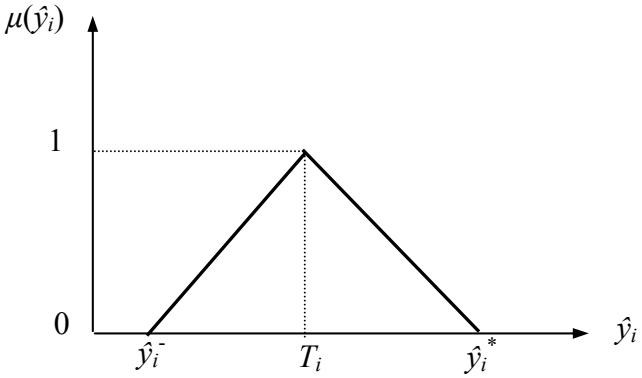

Figure 5. Membership function of $\mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x}))$: "the nominal the best" case

The problem formulation of (4) cannot be directly solved by the use of derivative-based methods due to unknown functional forms of $\hat{y}_i(\mathbf{x})$. Derivative-free methods such as genetic algorithm and simulated annealing are ideally suited for solving problems where derivative information is unavailable. Alternatively, we can approximate the derivatives with numerical methods. In this study, we will apply GA and a numerical method, respectively, to solve (4).

4. Solution procedures

In the previous section, we have formulated the multiple response optimization problem as an MODM. In this section, we suggest using two different algorithms, a genetic algorithm and a numerical method, to solve this MODM.

4.1. Genetic algorithm

Genetic algorithm first proposed by Holland [5] is a derivative-free stochastic optimization approach based on the concepts of biological evolutionary processes. GA encodes each point in a solution space into a binary bit string called a chromosome. Operations of chromosomes including selection, crossover, and mutation, are used to generate new chromosomes so as to explore the solution space. Each chromosome is evaluated by a fitness function. Such a fitness function corresponds to the objective function of the original problem. A great variety of genetic algorithms have been proposed in the literature. In this study, we will just use a basic form of GA. Nevertheless, it performs well as observed in our computation results. This genetic algorithm contains a roulette wheel selection, a single point crossover, and a random flipping mutation.

The fitness of chromosomes is determined via (4). It is not straightforward to determine the values of λ for a certain solution by using (4). Therefore, we rewrite (4) as

$$\begin{aligned} &\max \ \lambda \\ &\text{s.t.} \ \lambda = \min_{i=1,\ldots,m} \left\{ \mu_{\hat{y}_i}(\hat{y}_i(\mathbf{x})) \right\}, \\ &\mathbf{x} \in B. \end{aligned} \quad (8)$$

By employing the trained MANFIS, the formulation of (8) is presented in a network form in Figure 6, and λ can be directly read from the output end of this network. Genetic algorithm for solving the multiple response optimization problem has been formulated in our earlier paper [3]. In this paper, we further investigate its performance with computational experiments in Section 5.

4.2. Numerical method

A numerical method based on Lagrange relaxation for solving the multiple response optimization problem has been formulated in

[2]. Though this earlier method has the

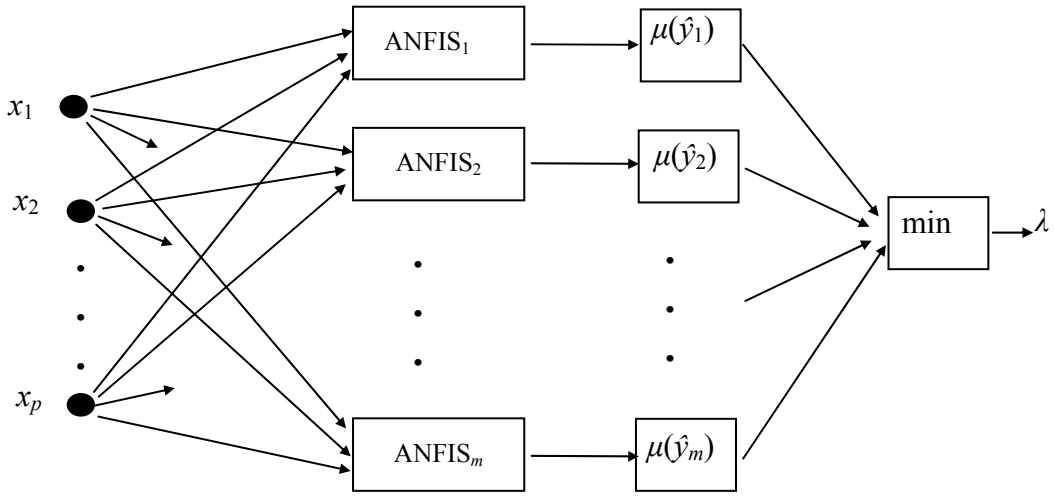

Figure 6. Network presentation of formulation (8)

advantage of providing an upper bound of the optimal solution, it is rather complicated. In this paper, a simpler numerical method, which directly solves the primal problem, is formulated as the follows.

Recalling the formulation of (8), to indicate λ being a function of $\mathbf{x}$ we denote it as $\lambda(\mathbf{x})$. The gradient of $\lambda(\mathbf{x})$ is defined as

$$\nabla\lambda(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial\lambda(\mathbf{x})}{\partial x_1} \\ \vdots \\ \dfrac{\partial\lambda(\mathbf{x})}{\partial x_p} \end{bmatrix}. \qquad (9)$$

By fixing the values of all $x_{k\neq j}$, and by giving a small increment Δx_j on x_j, the partial derivative $\dfrac{\partial\lambda(\mathbf{x})}{\partial x_j}$ can be approximated through

$$\frac{\partial\lambda(\mathbf{x})}{\partial x_j} \cong \frac{\lambda(x_1,\ldots,x_j+\triangle x_j,\ldots,x_p)-\lambda(x_1,\ldots,x_j,\ldots,x_p)}{\triangle x_j} \qquad (10)$$

After the gradient is determined, the maximization of $\lambda(\mathbf{x})$ can be done by an iterative manner through the updating of $\mathbf{x}$, subject to the feasible region constraint. The rule of this updating is

$$\mathbf{x}^{new} = \mathbf{x}^{old} + s\nabla\lambda(\mathbf{x}^{old}), \qquad (11)$$

where s is a step size. The steps of this numerical method are summarized below.

Step 0. Initialization:
set the iteration counter $r = 0$, the accuracy requirement τ, and the step size s; arbitrarily choose initial value $\mathbf{x}^0$ within the feasible region.

Step 1. Gradient calculation:
calculate the gradient of λ through Eq. (10) with $\mathbf{x}^r$.

Step 2. Updating of $\mathbf{x}$:
$\mathbf{x}^{r+1} = \mathbf{x}^r + s\nabla\lambda(\mathbf{x}^r)$.

Step 3. If $\lambda(\mathbf{x}^{r+1}) - \lambda(\mathbf{x}^r) \leq \tau$, stop;
otherwise, go to Step 4.

Step 4. Increase the iteration counter $r \leftarrow r + 1$.
Go to Step 1.

5. Computational comparison

To illustrate our approach, a chemical process optimization problem taken from Myers and Montgomery [12] is reproduced as the follows.

A chemical process has three controllable variables: reaction time (x_1), temperature (x_2), and percent catalyst (x_3); and its responses are percent conversion (y_1), and thermal activity (y_2). For this process, it is important to maximize y_1 while y_2 is held between 55 and 60 with a nominal target 57.5. Experiments are conducted with different setups of reaction time, temperature, and percent catalyst to collect data of this chemical process.

5.1. Modeling by MANFIS

MANFIS is employed to model the response surfaces of the above chemical process. The MANFIS for this problem has two output nodes corresponding to the two responses of this process, and hence this MANFIS consists of two independent ANFIS. The Fuzzy Toolbox in MATLAB software provides functions of constructing, editing and training of ANFIS. We use this software to construct two independent ANFIS networks and train them by the experimental data. The convergence behavior of one of the learning process is shown in Figure 7.

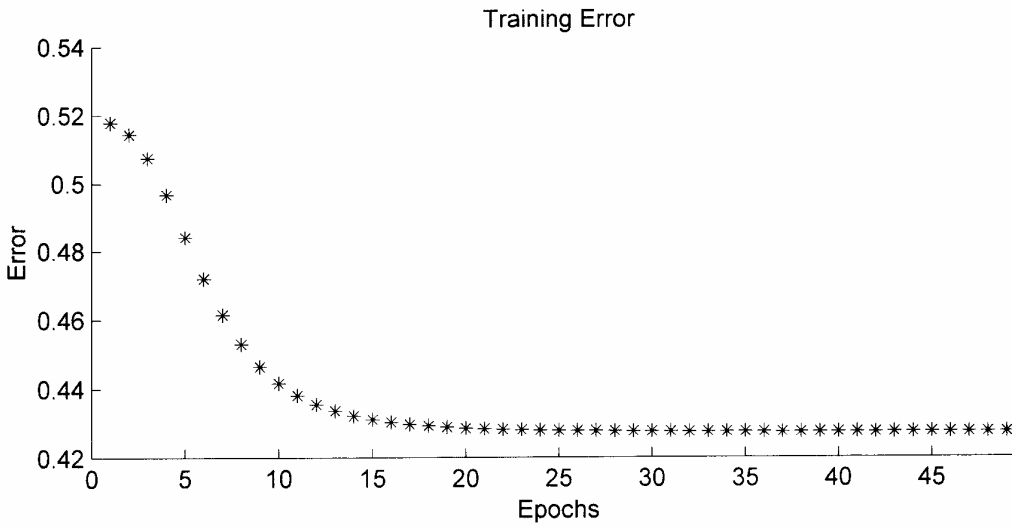

Figure 7. Convergence of the learning of y_2

After completing the training of MANFIS, the multiple response problem is solved by using the formulation of (8). Since the response y_1 belongs to the set of the-larger-the better, its membership function should take the form of (5); and the response y_2 has a nominal target, so it will take the membership function (7). In order to determine these membership functions, the maximum and thermal activity to be held between 55 and 60, therefore, it is reasonable to set 55 and 60 as minimum for individual response must be obtained. Maximum and minimum of responses can be obtained by formulating single objective programming problems for individual responses, and solving the problems with any derivative-free algorithm. Alternatively, they can also be subjectively determined according to users' judgment or their expectation. In our example, it is desired that the response of the minimum and maximum of this response, respectively. Similarly, the minimum and ma-

ximum of the response of percent conversion are set as 50 and 100, respectively. The possible ranges for x_1, x_2 and x_3 are set as [-2, 2].

5.2. Solving by GA

The genetic algorithm is implemented on the MATLAB platform and run on an IBM compatible PC with Pentium III-800 CPU. Ten trials are conducted, with the parameters in GA setting as: population size = 24, cross-over rate = 0.7, and mutation rate = 0.12. The results are listed in Table 1, in which, the second column is the time (in seconds) consumed by each trial to obtain the best solution, and the last two columns are the responses yielded by the best solutions. From Table 1 we can see that all the ten trials produce high quality solutions, i.e. all trials except the third trial obtain optimal solutions. Nevertheless, they usually consume a lot of computation times.

Table 1. Results of chemical process optimization by GA

Trial	Time	λ	Responses	
			y_1	y_2
1	313	0.999	100	57.5
2	585	0.998	100	57.5
3	1672	0.982	99	57.5
4	1086	0.997	100	57.5
5	415	0.996	100	57.5
6	1714	0.997	100	57.5
7	1157	0.999	100	57.5
8	539	0.999	100	57.5
9	1917	0.991	100	57.5
10	1497	0.996	100	57.5

5.3. Solving by the numerical method

The numerical method is also implemented on the MATLAB platform and run on the same machine. Ten trials are also carried out and their results are shown in Table 2. The starting points in these trials are arbitrarily chosen. We found that the quality of solutions obtained by this numerical method cannot compete with those obtained by GA, and the starting points critically affect the results. In number method should reach the optimal solution very fase. On the other hand, though particular, four (Trial 3, 6, 8 and 9) out of ten trials are failed because of the starting points are falling in a flat area of the response surface and hence no gradient can be found.

5.4. Combining GA and numerical method

Though the numerical method failed to produce high quality solution, it can fast solve the problem. If we can provide a starting point in the vicinity of the optimal solution, the GA usually takes a long time to find a high quality solution, in the first few generations.

Table 2. Results of chemical process optimization by the numerical method

Trial	Time	λ	Responses	
			y_1	y_2
1	126	0.763	88	56.7
2	134	0.763	88	56.9
3	-	-	-	-
4	217	0.967	98	57.4
5	224	0.762	88	56.9
6	-	-	-	-
7	85	0.763	88	56.9
8	-	-	-	-
9	-	-	-	-
10	209	0.763	88	56.9

Table 3. Results of chemical process optimization by the combined algorithm

Trial	Alg.	Time	λ	Responses	
				y_1	y_2
1	GA	83	0.78		
	NM	135	0.91	96	57.7
2	GA	84	0.96		
	NM	3	0.98	99	57.5
3	GA	86	0.79		
	NM	61	0.84	92	57.1
4	GA	89	0.91		
	NM	10	0.99	100	57.6
5	GA	83	0.89		
	NM	111	0.91	95	57.3
6	GA	83	0.95		
	NM	4	0.96	98	57.4
7	GA	83	0.97		
	NM	2	0.98	99	57.5
8	GA	83	0.88		
	NM	*			

* : no improvement

The idea is to combine these two algorithms together. That is, use GA to find a starting point for the numerical method. We have conducted 8 trials to justify this idea. In each trial, GA is run first for 20 generations to obtain a solution, and this solution will serve as a starting point for the numerical method. Computational results are shown in Table 3, in which, the second column is the algorithm used and NM denotes the numerical method. In Trial 1, 3 and 4, the numerical method provides significant improvement of the starting solutions, and among which, Trial 1 and 4 produce high quality solutions. Though Trial 2, 5, 6, and 7 do not significantly improve their starting solutions, they all produce high quality solutions. It is observed that the starting solutions of these trials are already in good shapes; and this may be why it is difficult for the NM to improve much on these starting solutions. The final trial found no improvement for its starting solution, possibly caused by the starting solution falling on a plateau of the response surface. Though the results in Table 3 show imperfection of the combined algorithm, we still consider the combination of GA and numerical method is promising for two reasons: 1) this combined algorithm consumes much less time than GA to find a satisfactory solution and, 2) it is possible to find a high quality solution in a moderate number of trials.

6. Application to a wire bonding problem

Wire bonding is a welding process, in which wire and pad surface are brought into intimate contact by using thin wire and a combination of heat, pressure and ultrasonic energy. Dynamic random access memory (DRAM) chips and most commodity chips in plastic packages are assembled by wire bonding. About 1.2-1.4 trillion wire interconnections are produced annually.

Wire bonding failures include bond off center, bond not sticking on die, wire breaking and so on. In a production environment, wire pull strength is usually monitored to minimize process drift. To achieve a stable performance of the wire bonding process, the operating variables such as bonding parameters need to be strictly regulated. Critical bonding parameters include bonding force, bonding time, and ultrasonic power. To find optimal setups of bonding parameters, traditionally, a serious of bonding tests is performed by varying bonding parameters to draw out the optimal bonding conditions1. Evaluation of wire pull strength is used to define the optimality of bonding parameters. In the evaluation, three sets of curves of wire pull strength versus ultrasonic power, bonding time, or bonding force can be obtained by varying one of these parameters while holding the other two constant at their optimum. In such an approach, the search is less efficient and frequently falls in a local optimum especially when the response surface is highly nonlinear.

To demonstrate the potential usage of our approach in real-world problems, the proposed approach in Section 4 is applied to the optimization of a wire bonding process in an IC packaging company in Taiwan. By fixing parameters of cut mode, heat, and looping of the wire bonder, and varying the parameters of bonding force, bonding time, and ultrasonic power, a design of experiment is conducted to collect data of the process. Each of the three variable parameters is set with three levels, and hence the experiment results in a combination of 27 trials of bonding tests. Each trial contains 100 replications, and the two concerning responses are average wire pull and its deviation, where average wire pull is a the-larger-the-better response, and deviation is a the-smaller-the-better response.

Firstly, MANFIS is employed to model the responses of this process. To ensure the generality of the MANFIS model, and to fully utilize the limited number of experimental data, an extreme cross-validation technique called leave one-out cross-validation [14] is used. The idea of cross-validation is to divide the sample data into a construction

sub-sample, which forms the training data set, and a validation sub-sample, which forms the test data set. The leave one-out cross-validation is to divide the sample size n into a training data set containing n-1 observations, and leave the rest single observation as the test datum. Such a technique considers the division of the observations in all n possible ways. The cross- validation criterion is defined as

$$CV(\mathbf{P}) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{k=1}^{m}(y_{ik} - \hat{y}_{ik}[O \setminus i])^2, \qquad (12)$$

where **P** is the set of critical factors that affect the accuracy of MANFIS, and $\hat{y}_{ik}[O \setminus i]$ is an estimate of y_{ik} and it is obtained from an MANFIS that is trained by the sample data excluding the i-th datum. The set of critical factors **P** contains only one factor, the number of nodes (in layer 1) associated with an input variable. To find the best setup of **P**, $CV(2)$, $CV(3)$, and $CV(4)$ is compared, and we found that $CV(3)$ is the minimum. With the result of cross-validation, the MANFIS for modeling the wire bonding process is constructed as: two independent ANFIS, and in each ANFIS there are 3 nodes associated with each input and hence resulting in 9 nodes in layer 1, 27 nodes in layer 2, 27 nodes in layer 3, and 27 nodes in layer 4.

The optimization of the wire bonding process is modeled by the formulation of (3), with a primary objective of maximizing the wire pull and a secondary objective of minimizing process variation. To construct membership functions for these two objectives we need to know their respect minimum and maximum as defined in Section 3. The minimum of wire pull is set as its specification (i.e. its minimal requirement), and the maximum of wire pull is determined according to past experience of running test on bonding. The minimum and maximum of the process variation are determined through a process performance index (Ppk). The definition of Ppk for one-sided specification (lower limit only) is

$$\text{Ppk} = \frac{\mu_p - LSL}{3\sigma_p}, \qquad (13)$$

where μ_p is the mean of the process, σ_p is the deviation of the process, and LSL is the lower limit of the process. Since the company is pursuing six-sigma process capability, we use this performance goal to determine the expected minimum of the process variation; that is, we set Ppk = 2 and induce the minimum of σ_p as (μ_p - LSL)/6. Furthermore, a company usually needs to reach a process capacity higher than four-sigma to satisfy most customers, and hence we can determine the maximum for δ_p as (μ_p - LSL)/3.99.

By employing the genetic algorithm to solve the wire bonding optimization problem based on the formulation of (8), we obtain a solution that is better than the company's current software can find.

7. Concluding remarks

This study used a neuro-fuzzy network, MANFIS, to model a multiple response system, and optimizes the system by a genetic algorithm and a numerical method respectively. MANFIS provides the advantage of modeling a nonlinear and complicated system without the need of finding suitable functional forms for the system, and its neural network learning ability also equips MANFIS with high efficiency in system modeling. A chemical process optimization problem is used to illustrate our approach. From the computational results, it is found that GA always finds process conditions that yield very satisfied responses, though it consumes much computational time. On the other hand, the numerical method is fast but its solution quality cannot compete with GA's. To improve performances of these two algorithms, we combine GA and the numerical method by running GA first for few generations to obtain a

starting solution for the numerical method. Computational results show that this combined algorithm is promising. The proposed approach of process optimization is applied to a wire-bonding problem in IC manufacturing.

References:

[1] Ames, A. E., Mattucci, N. S., MacDonald, G. Szonyi, and Hawkins, D. M. 1997. Quality loss functions for optimization across multiple response surfaces. *Journal of Quality Technology*, 29: 339-346.

[2] Cheng, C. B. 2000. Multi-response optimization based on a neuro-fuzzy system. *Neural Network World*, 10: 545-551.

[3] Cheng, C. B., Cheng, C. J., and Lee, E. S . 2002. Neuro-fuzzy and genetic algorithm in multiple response optimization. *Computers and Mathematics with Applications*, 44: 1503-1514.

[4] Derringer, G. and Suich, R. 1980. Simultaneous optimization of several response variables. *Journal of Quality Technology*, 12: 214-219.

[5] Holland, J. H. 1975. *"Adaptation in natural and artificial systems".* University of Michigan Press, Michigan.

[6] Jang, J. S. R. 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics,* 23: 665-684.

[7] Jang, J. S. R., Sun, C. T., and Mizutani, E. 1997. *"Neuro-Fuzzy and Soft Computing: a Computational Approach to Learning and Machine Intelligence".* Prentice-Hall. New Jersey.

[8] Khuri, A. I. and Conlon, M. 1981. Simultaneous optimization of multiple responses represented by polynomial regression functions. *Technometrics*, 23: 363-375.

[9] Kim, K. J. and Lin, D. 1998. Dual response surface optimization: a fuzzy modeling approach. *Journal of Quality Technology*, 30: 1-10.

[10] Lin, D. and Tu, W. 1995. Dual response surface optimization. *Journal of Quality Technology*, 27: 34-39.

[11] Myers, R. H. and Carter, W. H. 1973. Response surface techniques for dual response systems. *Technometrics*, 15: 301-317.

[12] Myers, R. H. and Montgomery, D. C. 1995. *"Response Surface Methodology".* John Wiley and Sons, Inc. New York.

[13] Pignatiello, J. J. Jr. 1993. Strategies for robust multiresponse quality engineering. *IIE Transactions*, 25: 5-15.

[14] Stone, M. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society.* Series B, 36: 111-147.

[15] Takagi, T. and Sugeno, M. 1985. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*,15: 116-132.

[16] Vining, G. G. A. 1998. Compromise approach to multiresponse optimization. *Journal of Quality Technology,* 30: 309-313.

[17] Zimmermann, H. J. 1978. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1: 45-55.