Robust Parameter Design by Neural networks and Genetic Algorithms

Chih-Hsien Chen[1] and Hsu-Hwa Chang[2,*]
[1]Marketing & Logistics Department
Lee-Ming Institute of Technology, Taipei, Taiwan
[2]Department of Business Administration
National Taipei College of Business, Taipei, Taiwan
*Email: hhchang@webmail.ntcb.edu.tw

Abstract

Taguchi's robust parameter design has been widely applied to a variety of quality engineering problems; however, it is unable to deal with dynamic multiresponse owing to the increasing complexity of the product or design process. This study incorporates desirability functions into a hybrid neural network/genetic algorithm approach to optimize the parameter design of dynamic multiresponses with continuous values of parameters. The objective is to find the optimal combination of control factors to simultaneously maximize the robustness of each response. The effectiveness of the propose approach is illustrated with a simulated example. The results of analysis reveal that the approach has higher performance than the traditional experimental design does.

1 Introduction

The Taguchi method is a traditional approach for robust experimental design that seeks to obtain the best combination of factor/level for the lowest societal cost while fulfilling customers' requirements. Over the past decade the Taguchi method has been widely applied to optimize the parameter design problems, which uses orthogonal array (OA) to arrange the experiments and employs signal-to-noise ratio (SNR) to evaluate the performance of the response of each experimental run. Nevertheless, Taguchi's method can only be used to resolve an optimal single response problem; it cannot be used to simultaneously optimize the multiresponse problem [1–4]. Unfortunately, in the real world, most customers consider more than one quality response problem, while selecting industrial products. In addition, the goals of the multiresponses often conflict with each other. A number of studies primarily focus on a multiresponse in a static system for manufactured products or processes that have been published [5–10]. Since most manufacturing processes are naturally dynamic systems, dynamic multiresponse problems may frequently be encountered in practice [4,11,12]. Hence, the parameter design problems containing dynamic multiresponses have increasingly received attention. Several researchers have begun to study this problem [12–15].

Considering a dynamic system with multiple responses, suppose that there are r output responses $Y=(y_1,y_2,\dots,y_r)$ which are determined by a set of control factor combination $\mathbf{X}$ and by a set of signal levels $M=(M_1,M_2,\dots,M_s)$. Figure 1 shows the Parameter Diagram of a dynamic system with multiresponse. The goal of the system is to determine the best settings of control factors so that the system's multiresponse have the least sensitivities to noise factors along the magnitude of the signal factor.

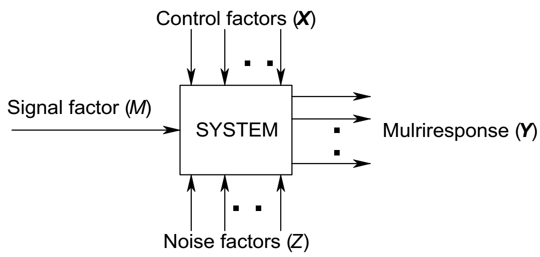

Figure 1. The Parameter Diagram of a dynamic multiresponse system

A dynamic system with multiresponse can be defined as:

$$y_{jk}=f_{jk}(M_k,\mathbf{X})+e_{jk}, \quad (1)$$

for $j=1,2,\dots,r$; $k=1,2,\dots,s$.

where f_{jk} denotes the response function between the control factors and the jth response at the kth level of signal factor; and e_{jk} is a random error.

For each dynamic response, it is assumed that a linear form exists between the response and the signal factor. The ideal function can be expressed as $y=\beta M+e$, where y denotes the response, M stands for the signal factor, β is the slope or system's sensitivity, and e represents the random error [16]. Further, dynamic systems can be classified into dynamic nominal-the-better (DNB), dynamic larger-the-better (DLB), and dynamic smaller-the-better (DSB) according to the desired type of response. Hence, the ideal target function is replaced as $y=\beta_t M+e$, where β_t is the desired target slope. For the response type DNB, DLB, and DSB, the value of the slope is $0<\beta_t<\infty$, $\beta_t=\infty$, and $\beta_t=0$, respectively.

In this work, we propose a novel optimization approach incorporating desirability functions into a hybrid neuron-genetic technique for resolving the dynamic multiresponse. The approach integrates neural networks (NN), exponential desirability functions and a genetic algorithm (GA) to model the system's response function and to optimize the parameter design. Using the proposed approach, the obtained optimal parameter settings can be any value within their upper and lower bounds.

2 Desirability Functions, NNs and GAs

For simultaneously optimizing the multiresponse problems, the most popular method is the exponential desirability function approach [17]. The desirability function transforms a predicted response to a scale-free value *d*, called desirability. It is a value between 0 and 1, and it increases as the desirability of the corresponding response increases. Many articles have used desirability functions to resolve the parameter design problems [7, 18]. By using this method, multiresponse can be converted into an OPI to evaluate a system's overall performance, which is suitable for applying to this study.

Recent works have discussed applying the method of integrating NNs and GAs to optimize the parameter settings of engineering designs [19-23]. A NN is used to construct the response function of a system; a GA is then applied to the network for searching the parameter settings with an optimal response. NNs are composed of processing elements and connections. Among several networks the supervised learning network named back-propagation neural (BPN) is most suitable for applying to parameter design because its ability of approximating any continuous mapping from the input patterns to the output patterns. BPN is a multi-layer network with learning ability. The nonlinear transfer function of sigmoid function is used between the connections of input layer, hidden layer, and output layer. Each layer is formed by several nodes and an additional bias node. BPN learning employs a gradient-descent algorithm to minimize the root of mean-square error (RMSE) between the target data and the predictions of the neural network. The training data set is initially collected to develop a BPN model. Applying a supervised learning rule, the data set is comprised of an input and an actual output (target). The gradient-descent learning algorithm enables a network to enhance its performance by self-learning. The training of a BPN involves three stages: the feedforward of the input training data, the calculation and back-propagation of the associated error, and the adjustment of the weights. While training the network model, the performance of the model is sensitive to various network structure choices and the parameter settings of learning rate and momentum coefficient. A common approach to obtain a well-trained network structure is to use the trial and error method, i.e., we can train several candidate networks that have a different number of hidden layers and nodes in each hidden layer, and then select the one with the smallest RMSE [24].

GA starts with an initial set of random solutions called a population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. A chromosome is a string type, which is organized by a sequence of the factors values for the problem. The individual sites on the chromosome where the parameter values are stored are called genes. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated by a fitness function. To create the next generation, GA applies a reproduction operator to select the candidate chromosomes from the present generation. The fitter chromosomes have a higher probability of being selected. And then, GA uses a crossover operator and mutation operator to create the offspring. The process continues until a desirable solution is obtained or a predetermined generation size is reached [25].

3 Proposed Approach

The proposed approach consists of three stages. First, experimental data are collected to train a BPN to represent the response function model of a dynamic multiresponse system, f_{jk}, which is capable of predicting the corresponding multiresponse by giving any factor combinations within the feasible solution space. The second stage involves using desirability functions to evaluate the performance measures of the predicted multiresponse for the three types of dynamic systems. The performance measures are then integrated into an OPI value to represent the total performance of a specific factor combination. Finally, a GA is utilized to obtain the optimal OPI value and the corresponding factor combination. Figure 2 shows the flowchart of the approach.

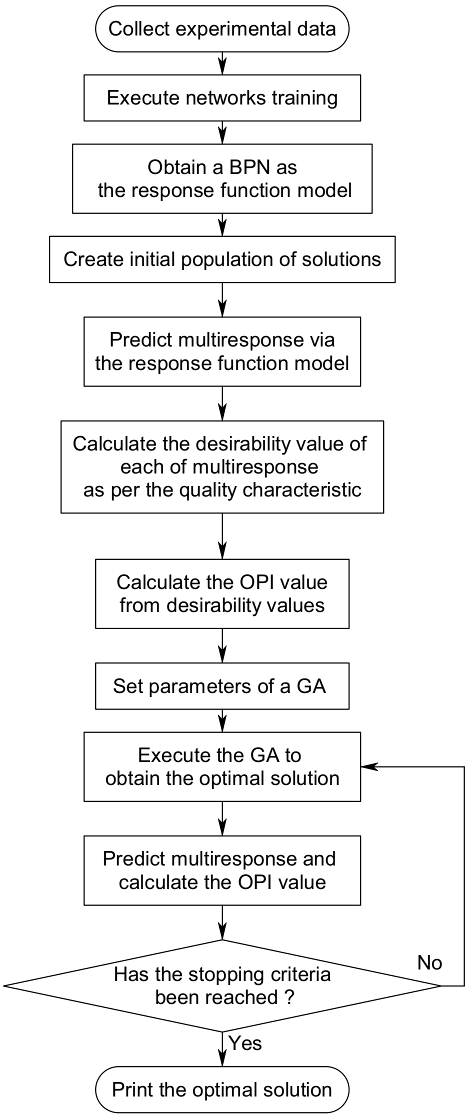

Figure 2. The proposed approach

3.1 Response Modeling

This stage uses a BPN to model the response function, which builds the relationship function between the multiresponse and the parameters of a system. The input data are assigned as the control factor values and the signal values; the output data are the multiresponse. A well-trained BPN represents the system's response function model, i.e., $\hat{y}_{jk} = f_{jk}(M_k, \mathbf{X})$. The processes of this stage are described as follows:

Step 1. Collect the training and testing patterns for input and output layers from the experimental data.
Step 2. Select several candidates of network structures for training.
Step 3. Set learning rate, momentum coefficient and execution iterations L.
Step 4. For each network structure, Steps 5—8 are executed L times.
Step 5. Initialize randomly weights between layers.
Step 6. Apply the sigmoid function $f = 1/(1+e^{-x})$ to predict the outputs.
Step 7. Calculate the error between the predicted output and the target output.
Step 8. Adjust the weights of the network.
Step 9. Choose the best one from the several trained networks as the system's response function model. The performance evaluation criterion for the network training is the RMSE.

3.2 Evaluating the OPI Value

This works modifies the exponential desirability functions introduced by Harrington [10] for applying to the dynamic multiresponse system. For the three types of dynamic responses, the desirability value of each type of the predicted multiresponse can be developed as:

DNB: $d^{DNB} = \exp(-Z^{DNB})$, (2)

where $Z^{DNB} = \frac{1}{s}\sum_{k=1}^{s}\left|\frac{2\hat{y}_{jk} - (y_{jk}^{\max} + y_{jk}^{\min})}{y_{jk}^{\max} - y_{jk}^{\min}}\right|$;

DLB: $d^{DLB} = \exp(-(\exp(-Z^{DLB})))$, (3)

where $Z^{DLB} = \frac{1}{s}\sum_{k=1}^{s}\frac{\hat{y}_{jk} - y_{jk}^{\min}}{y_{jk}^{\min}}$;

DSB: $d^{DSB} = \exp(-(1+Z^{DSB}))$, (4)

where $Z^{DSB} = \frac{1}{s}\sum_{k=1}^{s}\frac{\hat{y}_{jk} - y_{jk}^{\max}}{y_{jk}^{\max}}$.

For Equations (2)—(4), the bounds $y_{jk}^{\max}$ and $y_{jk}^{\min}$ represent the upper specification limit (USL) and lower specification limit (LSL) for the jth response at the kth signal level. To evaluate the overall performance of the multiresponse, we can formulate the fitness function by integrating the multiple desirability values into an OPI value as Equation (5):

$$OPI = \sqrt[r]{\prod_{j=1}^{r} d_j}, \quad (5)$$

where d_j denotes the desirability value for the *j*th response; $j = 1, 2, \ldots, r$.

3.3 Optimizing

This stage involves performing a GA to optimize the OPI value for obtaining the optimal multiresponse and the corresponding combination values of the control factors from the possible solution space. Herein, a possible solution represents a chromosome; an OPI stands for the fitness value of the GA. Genes in the chromosome are formed by the values of the control factor and the values of the signal factor. The parameter bounds and the precision are determined according to the characteristics of the systems. The operational steps are given as follows:

Step 1. Set population size, crossover rate P_c, and mutation rate P_M. Initialize a random population of string of size *l*. Choose a maximum allowable generation number $t_{\max}$. Set $t = 0$.

Step 2. Calculate the predicted multiresponse by inputting the control factor values and the signal values to the response function model f_{jk}, i.e., the trained BPN in Stage 1.

Step 3. Evaluate the OPI value through Equations (2)—(5).

Step 4. If $t > t_{\max}$ then terminate.

Step 5. Perform reproduction on the population.

Step 6. Perform crossover on pair of string with probability P_c.

Step 7. Perform mutation on strings with probability P_M.

Step 8. Evaluate the OPI values of strings in the new population. Set $t = t + 1$ and go to Step 2.

Step 9. Obtain the optimal combination values of control factor and the corresponding multiresponse through the response function model f_{jk}.

4 Illustrative Example

4.1 Description of the Example

The proposed approach is illustrated with a simulated example of a dynamic system containing multiresponse. Suppose there are three responses named y_1, y_2 and y_3 to be simultaneously optimized. The quality characteristics of the y_1, y_2 and y_3 are DLB, DNB and DSB, respectively. Simulated experimental data are obtained based on the Monte Carlo simulation and the procedures of Park and Yum [26]. Six control factors named *A*, *B*, *C*, *D*, *E* and *F*, each at three levels (i.e. level 1, 2, and 3), are respectively allocated to columns 3—8 in the order as they appear in orthogonal array L_{18}. The signal factor has three levels named M_1, M_2 and M_3, the corresponding values are 0.1, 0.2 and 0.3, respectively. The specification limitations of the three responses at each signal level are listed in Table 1. Experiments are conducted with two replicates at each of the control factor settings. The results of the experiments are given in Table 2. To evaluate the performance of the multiresponse of the experiments, we calculate the desirability and the OPI value of each experimental run, which are listed in Table 3. Table 3 shows that the repetition 1 of run 3 has the largest OPI = 0.66130 and the repetition 2 of run 6 has largest OPI = 0.69863.

4.2 Constructing the Response Function Model

The response model of the system is constructed by a BPN. This BPN is trained by assigning the (control factor values, signal value) /multiresponse as the inputs/outputs of the network. For building a well-trained network, we randomly select 92 training patterns and 16 testing patterns from Table 2. Table 4 lists several options of the network architecture; in addition, the structure 7-14-3 is selected to obtain a better performance. This study makes use of the neural network software package Qnet® (http://www.qnetv2k.com).

Table 1. The specifications of the three responses at each signal level

Responses		y_1			y_2			y_3		
Type		DLB			DNB			DSB		
Signal values		0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
Bounds	USL	N/A	N/A	N/A	1.4	2.8	4.2	28	56	84
	LSL	4.8	9.6	14.4	0.6	1.2	1.6	N/A	N/A	N/A

Table 2. The experimental data

No.	Responses																	
	y_1						y_2						y_3					
M	M_1		M_2		M_3		M_1		M_2		M_3		M_1		M_2		M_3	
Rep.	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	7.80	8.13	14.22	14.92	25.96	28.84	0.98	1.09	1.63	1.42	2.79	2.53	15.00	16.56	32.24	41.81	48.25	83.92
2	8.63	7.53	17.01	16.52	27.13	31.76	1.02	1.05	2.22	1.73	3.14	3.44	16.00	16.30	42.91	29.17	57.08	41.51
3	8.12	7.28	16.65	15.84	25.98	26.05	1.05	0.94	2.17	2.15	2.90	2.92	23.00	23.29	40.63	48.37	34.52	57.33
4	8.18	8.07	18.29	15.92	25.34	20.76	0.68	0.72	1.46	1.50	2.19	2.26	25.00	15.98	37.74	41.75	59.79	47.14
5	7.04	7.58	13.11	16.53	27.66	22.89	1.14	1.23	2.64	2.27	3.44	3.98	18.00	14.40	23.80	44.36	41.62	43.45
6	8.32	9.79	16.80	14.74	26.55	26.82	1.00	0.96	2.49	1.97	3.36	2.95	26.00	10.28	40.45	30.69	23.84	67.64
7	8.02	8.30	14.46	15.42	25.74	23.10	1.22	1.20	2.29	2.39	3.18	3.29	28.00	19.68	40.57	50.66	61.05	72.99
8	6.36	8.24	18.23	17.48	20.24	28.28	0.73	0.86	1.43	2.13	2.11	2.18	12.00	26.70	31.01	32.74	82.76	66.55
9	5.93	8.65	16.51	13.43	22.36	19.92	1.12	0.91	1.92	1.77	2.52	2.98	17.00	19.78	49.92	28.39	56.18	52.64
10	8.56	8.88	17.57	19.17	25.73	23.20	0.80	0.75	1.45	1.62	2.36	2.40	21.00	28.16	39.08	47.59	71.62	83.71
11	7.61	9.85	17.34	16.31	27.06	28.60	0.92	1.23	2.55	2.54	3.95	3.47	20.00	16.24	43.19	28.68	60.13	70.66
12	7.88	8.07	16.89	12.55	22.98	24.26	1.08	1.05	2.28	2.22	3.32	3.23	18.00	11.38	46.14	22.51	66.97	65.73
13	8.73	6.82	18.22	15.64	25.64	20.26	0.95	0.99	2.00	2.00	2.94	2.93	26.00	22.32	64.67	40.40	94.98	58.26
14	7.97	9.72	16.72	11.98	23.27	23.10	1.17	1.14	1.95	2.35	3.91	3.58	16.00	23.16	24.82	44.13	51.38	63.52
15	9.16	8.77	16.72	15.86	24.97	30.30	0.85	0.79	1.42	1.75	2.33	2.34	14.00	12.88	40.57	33.27	33.99	60.82
16	9.32	8.71	14.86	15.67	21.87	28.43	1.05	1.10	2.01	2.26	3.29	3.02	22.00	15.90	51.58	43.90	75.55	86.65
17	8.32	6.91	16.03	14.10	22.70	18.87	0.80	0.85	2.07	1.99	2.71	2.46	23.00	20.34	42.91	32.95	36.92	64.79
18	8.71	6.37	14.87	18.74	31.61	22.69	1.14	0.98	1.92	1.58	3.57	2.97	19.00	12.43	37.70	38.89	69.16	55.98

Table 3. The desirability and *OPI* values of each experimental run

No	Repetition 1				Repetition 2			
	d_1	d_2	d_3	*OPI*	d_1	d_2	d_3	*OPI*
1	0.58906	0.81952	0.57009	0.65046	0.62359	0.66263	0.45885	0.57448
2	0.64317	0.84377	0.51048	0.65189	0.64706	0.74630	0.58719	0.65697
3	0.62157	0.89360	0.52067	**0.66130*	0.59595	0.88902	0.45263	0.62128
4	0.63509	0.50981	0.46788	0.53308	0.57565	0.54565	0.53482	0.55177
5	0.57266	0.59345	0.59386	0.58657	0.58714	0.55928	0.54449	0.56336
6	0.63104	0.72461	0.52471	0.62138	0.64162	0.94303	0.56356	**0.69863*
7	0.59480	0.68663	0.44171	0.56504	0.59222	0.65105	0.43802	0.55275
8	0.55939	0.51424	0.51899	0.53049	0.64765	0.70085	0.45986	0.59320
9	0.54611	0.79392	0.48555	0.59488	0.55448	0.82584	0.54153	0.62825
10	0.63808	0.58608	0.46448	0.55796	0.64356	0.60964	0.38647	0.53325
11	0.62556	0.56832	0.48008	0.55470	0.66927	0.56960	0.52495	0.58492
12	0.59812	0.74750	0.47016	0.59458	0.56411	0.80416	0.58843	0.64387
13	0.64726	0.94940	0.34253	0.59485	0.54048	0.98410	0.47836	0.63366
14	0.60029	0.65608	0.58152	0.61182	0.58573	0.64606	0.45363	0.55576
15	0.63652	0.59880	0.58098	0.60499	0.65528	0.65523	0.55282	0.61916
16	0.59930	0.86430	0.41948	0.60118	0.64007	0.80056	0.45183	0.61404
17	0.59634	0.78306	0.50878	0.61935	0.51368	0.78506	0.49888	0.58596
18	0.65299	0.72485	0.48430	0.61201	0.58340	0.81089	0.54793	0.63760

Table 4. The candidate network structures

Structure	RMSE	
	Training	Testing
7-10-3	0.0484	0.0796
7-11-3	0.0478	0.0801
7-12-3	0.0489	0.0785
7-13-3	0.0464	0.0777
**7-14-3*	*0.0479*	*0.0776*
7-15-3	0.0458	0.0794
7-16-3	0.0444	0.0867
7-17-3	0.0443	0.0812
7-18-3	0.0468	0.0805
7-19-3	0.0459	0.0811

Note: Learning rate is set as between 0.01 and 0.3; momentum coefficient is 0.80; number of iterations is 10,000.

4.3 Performing the GA

In this stage, a GA is performed to obtain the optimal OPI value within the feasible solution space of the system. The values of the six control factors are set as continuous and fall in the range between 1 and 3. The operational conditions of the GA are set as: (number of generation $t_{\max}$: 3000), (population size l: 80), (crossover rate P_c: 0.80), and (mutation rate P_M: 0.085). The GA program is executed over 20 runs to obtain the optimal OPI value 0.753553 and the corresponding values of factor combination. Through the response function model, the predicted responses at the optimal values of factor combination can be obtained. Table 5 lists the predicted responses at the optimal values of factor combination. Table 6 lists the optimal OPI value and the corresponding values of factor combination. For the purpose of benchmarking, this study conducted a comparison between full factor/level combinations and the proposed approach. Table 6 also lists the best OPI value 0.745636 of full factor/level combinations. It is worthy to notice that the obtained factor values of the best combinations are restricted to the experimental levels while using full factor/level combinations to resolve parameter design. Table 6 reveals the proposed approach outperforms the traditional experimental design in terms of the OPI value.

Table 5. The predicted responses at the optimal values of factor combination

Responses	y_1			y_2			y_3		
Signal values	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
Predicted multiresponse	11.51	19.89	26.85	1.18	1.99	2.95	5.22	17.38	47.08
d value	0.719104			0.846602			0.702863		

Table 6. The optimal values of factor combination

Method	optimal values						OPI value
	A	B	C	D	E	F	
Full combinations	2	3	3	1	2	1	0.745636
Proposed approach	1.85	2.99	2.97	1.00	1.86	1.00	0.753553

5 Discussion and Conclusion

Parameter design is a critical phase in developing new products because it mostly determines the total production cost and quality. However, parameter design problems are complicated because nonlinear relationships and interactions may occur among parameter, particularly in multiresponse which have different goals. A hybrid approach that incorporates exponential desirability functions into a neuron-genetic technique is proposed to optimize dynamic multiresponse systems and has been demonstrated with an illustrative example. Using the proposed response modeling approach, we expand the feasible combination space to infinity, unlike the traditional experimental design that can only find the best one from full combinations of experimental factor/level. For the illustrative example, the best combination will be selected from the $3^6 = 729$ combinations if a traditional experimental design is employed. By using the proposed approach, the obtained combination is formed by any values within the control factors' upper and lower bounds. Since the proposed approach can get much more feasible combination space than the traditional experimental design, it increases the probability of obtaining the optimal combination. Furthermore, the proposed approach unlike Taguchi method dose not use adjustment factors to optimize parameter design, which cannot be guaranteed to exist in practice. Also, the results of analysis reveal that by applying the approach the obtained factor values of optimal parameter settings are not limited to the discrete values of the experimental levels, and have higher performance than the full combination does in terms of the OPI value. The proposed approach provides a generalized solution for parameter design and can be applied to diverse industrial fields. Moreover, through appropriate modification, the approach can be reduced to deal with most of the situations that practitioners may encounter, including static multiresponse, simple dynamic systems, and general static problems.

Acknowledgements

This research project was sponsored by the National Science Council of Taiwan under Grant No. NSC94-2416-H-141-003.

References

[1] Taguchi, G., Chowdhury, S. and Taguchi, S. (2000). *Robust Design Engineering*, McGraw-Hill.

[2] Maghsoodloo, S., Ozdemir, G., Jordan, V. and Huang, C.H. (2004). Strengths and limitations of Taguchi's contributions to quality, manufacturing, and process engineering, *Journal of Manufacturing systems*, **23**(2): 73-126.

[3] Robinson, T.J., Borror, C.M., and Myers, R.H. (2004). Robust parameter design: a review, *Quality and Reliability Engineering International*, **20**: 81-101.

[4] Zang, C., Friswell, M.I., and Mottershead, J.E. (2005). A review of robust optimal design and its application in dynamics, *Computers and structures*, **83**: 315-326.

[5] Liao, H.C. (2005). Using N-D method to solve multi-response problem in Taguchi, *Journal of Intelligent Manufacturing*, **16**: 331-347.

[6] Liao, H.C. (2006). Multi-response optimization using weighted principle component, *International Journal of Advanced Manufacturing Technology*, **27**: 720-725.

[7] Wu, F.C. (2005). Optimization of correlated multiple quality characteristics using desirability function, *Quality Engineering*, **17**: 119-126.

[8] Antony, J., Anand, R.B., Kumar, M. and Tiwari, M.K. (2006). Multiple response optimization using Taguchi methodology and neuron-fuzzy based model, *Journal of Manufacturing Technology Management*, **17**(7): 908-925.

[9] Liao, H.C., Chang, H.H. and Hsu, C.M. (2006). Using canonical correlation to optimize Taguchi's multiresponse problem, *Concurrent Engineering: Research and Application*, **14**(2): 141-149.

[10] Liao, H.C. and Chen, Y.K. (2002). Optimizing multi-response problem in the Taguchi method by DEA based ranking method, *International Journal of Quality and Reliability Management,* **92**: 241-254.

[11] Xydas, N., Tsi, D., Gurevich, V., Krichever, M. and Kao, I. (2005). Dynamic Taguchi methods and parameter design as applied in barcode scanning and scanners, *Concurrent Engineering: Research and Applications*, **13**(1): 69-80.

[12] Tong, L.I., Wang, C.H., Chen, C.C., Chen, C.T. (2004). Dynamic multiple responses by ideal function analysis, *European Journal of Operational Research*, **156**: 433-444.

[13] Wu, F.C., Yeh, C.H. (2005) Robust design of multiple dynamic quality characteristics, *International Journal of Advanced Manufacturing Technology*, **25**: 579-588.

[14] Hsieh, K.L., Tong, L.I., Chiu, H.P., Yeh, H. Y. (2005). Optimization of a multi-response problem in Taguchi's dynamic system, *Computers and Industrial Engineering*, **49**: 556-571.

[15] Wang, C.H., Tong, L.I. (2005). Optimization of dynamic multi-response problems using grey multiple attribute decision making, *Quality Engineering*, **17**: 1-9.

[16] Wu, Y. and Wu, A. (2000). *Taguchi Methods for Robust Design*, ASME Press.

[17] Harrington, E.C. (1965). The desirability function, *Industrial Quality Control*, **21**: 494-498.

[18] Castillo, E.D., Montgomery, D.C., McCarville, D.R. (1996). Modified desirability functions for multiple response optimization, *Journal of Quality Technology*, **28**: 337-345.

[19] Su, C.T., Chiu, C.C. and Chang, H.H. (2000). Optimal parameter design via neural network

and genetic algorithm, *International Journal of Industrial Engineering*, **7**(3): 224-231.

[20] Cook, D.F., Ragsdale, C.T., Major, R.L. (2000). Combining a neural network with a genetic algorithm for process parameter optimization, *Engineering Application of Artificial Intelligence*, **13**: 391-396.

[21] Chow, T.T., Zhang, G.Q., Lin, Z., Song, C.L. (2002). Global optimization of absorption chiller system by genetic algorithm and neural network, *Energy and Building*, **34**: 103-109.

[22] Öztürk, M., Yildiz, A.R., Kaya, N. and Öztürk, F. (2006). Neuro-genetic design optimization framework to support the integrated robust design optimization process in CE, *Concurrent Engineering: Research and Applications*, **14**(1): 5-16.

[23] Aijun, L.A., Hejun, L., Kezhi, L., Zhengbing, G. (2004). Applications of neural network and genetic algorithms to CVI processes in carbon/carbon composites, *Acta Materialia*, **52**: 299-305.

[24] Rao, M.A., Srinivas, J. (2003). *Neural Networks: Algorithm and applications*, Alpha Science International Ltd.

[25] Gen, M., and Cheng, R. (1997). Constrained optimization problems, *Genetic Algorithms and Engineering Design*, 42-96, John Wiley & Sons.

[26] Park, Y.G. and Yum, B.J. (2003). Development of performance measures for dynamic parameter design problems, *International Journal of Manufacturing Technology and Management*, **5**(1/2): 91-104.