Int J Adv Manuf Technol (2009) 40:1227–1238
DOI 10.1007/s00170-008-1423-7

ORIGINAL ARTICLE

An artificial neural network approach to multiple-response optimization

R. Noorossana · Sam Davanloo Tajbakhsh · A. Saghaei

Received: 5 August 2007 / Accepted: 28 January 2008 / Published online: 3 April 2008

Abstract In many manufacturing cases, engineers are required to optimize a number of responses simultaneously. A common approach for the optimization of multiple-response problems begins with using polynomial regression models to estimate the relationships between responses and control factors. Then, a technique for combining different response functions into a single scalar, such as a desirability function, is employed and, finally, an optimization method is used to find the best settings for the control factors. However, in certain cases, relationships between responses and control factors are far too complex to be efficiently estimated by polynomial regression models. In addition, in many manufacturing cases, engineers encounter qualitative responses, which cannot be easily stated in the form of numbers. An alternative approach proposed in this paper is to use an artificial neural network (ANN) to estimate the quantitative and qualitative response functions. In the optimization phase, a genetic algorithm (GA) is considered in conjunction with an unconstrained desirability function to determine the optimal settings for the control factors. Two manufacturing examples in which engineers were asked to optimize multiple responses from the semiconductor and textile industries are included in this article. The results indicate the strength of the proposed approach in the optimization of multiple-response problems.

Keywords Design of experiment · Response surface methodology · Artificial neural networks · Genetic algorithm, multiple layer perceptron · Radial basis function · Mean square error

1 Introduction

In today's highly competitive market, companies are impelled to constantly improve the quality of their products. The design of experiments (DOE) is an effective quality-improvement method recommended by most practitioners to optimize the performance of manufacturing processes. Experimental design methods help us to investigate the effects of control and noise factors on one or more responses of interest. Control factors, such as temperature and pressure, are those factors whose values can be controlled during manufacturing operations, while noise factors, such as ambient temperature or humidity, are factors whose values cannot be held constant in real-life situations. The goal of DOE is to determine the optimum settings for the control factors so that the product quality characteristic or response of interest attains its target with minimum variation. Such a product is commonly referred to as a robust product. However, in most cases, the evaluation of products or processes can involve the simultaneous study of several quality characteristics, with each one having its own relative importance to customers. This problem is commonly referred to as the multiple-response optimization problem. The ultimate goal of this optimization problem is

R. Noorossana · S. Davanloo Tajbakhsh (✉)
Industrial Engineering Department,
Iran University of Science and Technology,
Narmak, University Avenue,
Tehran 16846-13114, Iran
e-mail: s_tajbakhsh@ind.iust.ac.ir

A. Saghaei
Industrial Engineering Department,
Azad University, Research and Science Campus,
Tehran, Iran

to determine the settings for the control factors which lead to the best combination of the responses. More detailed discussions can be found in Montgomery [12].

2 Literature review

Different methods have been proposed in the literature for the optimization of multiple-response problems. Ortiz et al. [17] classify the existing methods into three basic categories. The performance of each method depends on the size and complexity of the problem. The first category consists of overlaying the contour plots of each response and finding the region of interest in which different responses are satisfied. The main problem with this approach is that it cannot identify the most dominant solution. Myers and-Montgomery [15] state that this approach is effective when only a few control factors are involved. The second category discussed by Ortiz et al. [17] consists of approaches which can be used to formulate the problem in the form of a constrained optimization problem. Kim et al. [10] refer to this category as priority-based approaches. The approaches in this category use the most important response to the decision maker as the objective function and the rest of the responses are considered as constraints. This approach is one of the basic approaches used in the multiple-objective decision-making problem. An example to such approaches can be found in the literature of response surface methodology (RSM) [3, 9]. According to Kim et al. [10], the main disadvantage of these approaches is that they do not conform to the basic idea of multi-response surface optimization to simultaneously consider all of the responses. Furthermore, the selection of a response as an objective function may not be easy in all cases.

The third and the most general category consists of approaches which use the following three steps to find the most dominant solution. First, they build models for the responses of interest. Second, they use an approach to combine the models into a single-value scalar. Finally, they optimize a single-value model by means of an optimization method. Ordinary least squares (OLS) is the most common approach proposed for accomplishing the first step, i.e., model building. However, Shah et al. [19] used seemingly unrelated regressions (SUR) to estimate model parameters. This method is very useful when the response variables in multiple-response problems are correlated. Regarding the second step, different techniques, including desirability functions [4–6], distance functions [8], squared error loss functions [18, 22], and proportion of conformance [2], have been proposed in the literature for combining multiple-response models into a single scalar. The desirability function methods are easy to understand and provide flexibility in weighting individual responses. However, the advantage of the other three techniques is their ability to consider the correlation that may exist among the responses. The optimization method in the third step of the above procedure depends on the properties of a single-value response surface model. Search techniques, such as the Nelder-Mead simplex [16] and Hooke-Jeeves [5], are commonly used for nondifferentiable response surface functions. However, for differentiable objective functions, one can consider gradient-based methods, such as generalized reduced gradient (GRG) [4]. Heuristic search procedures, such as genetic algorithms (GA), simulated annealing (SA), or tabu search (TS), could be applied to highly nonlinear, complex, and badly behaved surfaces [17].

As stated above, the third category is the most promising category developed so far to address the problems in multiple-response optimization (MRO). However, the main problem in this category arises when the mean square error (MSE) of the regression models are high, which is known as the poor quality of description [10]. This will happen in two situations. First, the main assumption for the independency of input variables is violated. Second, the relationship between responses and control factors are too complex, such that regression multipliers can not be estimated precisely [14]. In these situations, the final solution would be flawed. In this article, we have tried to develop a new approach based on an artificial neural network (ANN) first for problems in which regression models yield high MSE values. The use of an ANN has helped us first in detecting the significant control factors of each response and then to estimate the relation between that response and its significant control factors.

Another problem which often occurs in real-world applications is that users encounter qualitative responses. Qualitative responses are those quality characteristics which cannot be conveniently represented numerically. In many cases, we usually classify each item inspected as either "conforming" or "nonconforming" to the specifications on that quality characteristic. Quality characteristics of this type are called attributes ([13], Chap. 6). Some examples of quality characteristics that are attributes are the occurrence of warped automobile engine connecting rods in a day's production and the proportion nonfunctional semiconductor chips in a production run. In many other cases, the nature of a quality characteristic is such that it cannot be defined numerically. For instance, when the quality characteristic is a quality of a bond, an expert can only express his/her idea from the bond in the form of lingual expressions. Generally, traditional methods developed so far, use the Likert scale to transform qualitative values into quantitative values using the experience of experts. Tong and Hsieh [21] mentioned that their neural networks approach is not capable of handling problems with qualitative responses and it is an area for further research. Generally, there has not been a productive approach developed so far for dealing with problems including qualitative responses in the context of response surface methodology (which is a methodology for optimization). In this paper, a procedure based on fuzzy logic is considered to address the problem with qualitative responses.

The rest of the paper is structured as follows. Our proposed approach is discussed in detail in Sect. 3. In Sect. 4, the performance of the proposed approach is evaluated through two numerical examples. Our concluding remarks are exposed in the final section.

3 The proposed approach

ANNs have been widely used for function approximation. A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. ANNs are mainly used for function approximation and pattern recognition. Depending on which type of ANN we use, there are different parameters to tune, but the concept that they all share in common is that they all need to be trained. Usually, examples are used to train the neural network. Each example consists of an input–output pair: an input signal and its corresponding desired response for the neural network. Thus, a set of examples represents knowledge about the environment of interest [7]. Given such a set of examples, the design of a neural network may proceed as follows. First, an appropriate architecture is selected for the neural network. Second, a subset of examples is used to train the network by means of a suitable algorithm (learning). Third, the performance of trained network is tested with data that has not been used in training (generalization).

The proposed approach follows basically the same three-step procedure similar to the third category mentioned before. It utilizes neural networks to estimate the relation between control factors as inputs and responses as outputs at the first phase, unconstrained desirability functions combined with penalty functions (developed by Ortiz et al. [17]) at the second phase, and, finally, a GA as a potent optimization tool at the optimization phase. This procedure enables us to take advantage of neural networks' capabilities in function approximation, the potential of desirability functions in weighting individual responses, and also the aptitude of GAs in optimizing highly nonlinear, complex, and badly behaved functions. In situations where we have qualitative responses, our approach follows a two-step preprocessing procedure in which lingual expression will be transformed to vectors, so that they can be analyzed by neural networks. Figure 1 depicts a flow chart of the proposed approach.

3.1 Design of experiment

As it is depicted in Fig. 1, the proposed approach starts with designing an experiment. Experimental design helps us to

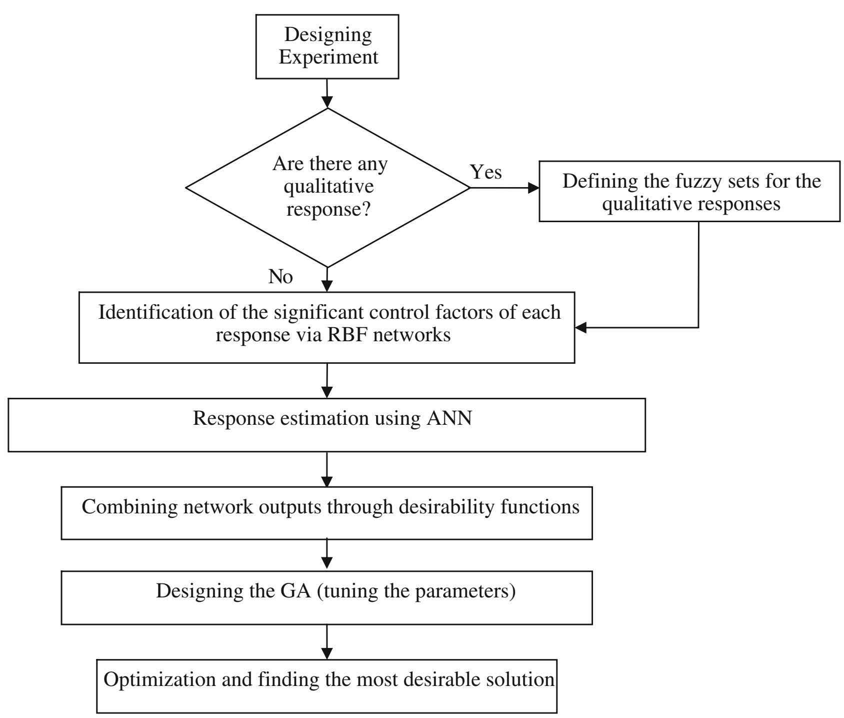

Fig. 1 Flow chart of the proposed approach

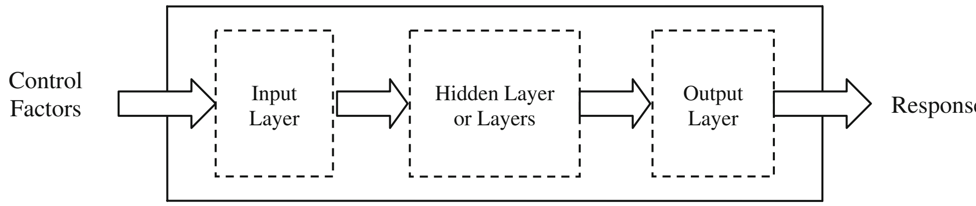

Fig. 2 Topology of neural networks

collect the required data for training the neural networks. RSM designs such as central composite (CCD) or Box-Behnken designs, due to their ability to provide the required information by covering the experimental space more thoroughly, are usually considered as effective designs for collecting the required data. Hence, such designs help the neural networks to approximate the process function more precisely.

3.2 Qualitative responses

For qualitative responses in the form of lingual expressions such as "Very Good," "Good," "Medium," etc., fuzzy sets are defined. In defining fuzzy sets, the implications of experts are commonly considered. This is one of the most important steps in fuzzifying a qualitative variable. The fuzzification of a qualitative response means to define different membership functions for different qualitative values of that response on a predefined domain [20].

ANNs are unable to process fuzzy sets in the form of membership functions. Hence, they are fed to the neural networks in the form of vectors. For this reason, the domain of fuzzy sets is divided to n equal intervals and in n+1 result points; the degrees of membership are represented by an n+1 element vector [11]. The n+1 element vector is an approximation to the membership function. It is obvious that increasing the value of n will result in a better approximation. Determining an appropriate value for n depends on the membership function's nature.

3.3 Application of an ANN for response estimation

At this phase, a neural network would be trained for each response to approximate its relation with control factors. Thus, the number of trained neural networks is equal to the number of responses. The inputs to these networks are control factors and the outputs are responses. Hence, the output layer of the networks for quantitative responses have a single neuron and the output layer of the networks for qualitative responses have n+1 neurons. Figure 2 illustrates the networks' topology. To avoid network memorization, a subset of 10–15% of the total data would be selected randomly as the *test data* and the rest is considered as the *training data*.

3.3.1 Identification of significant control factors for each response

Before training the networks, significant factors for each response have to be identified in the form of a subset of all of the control factors yielding the minimum MSE. Radial basis function (RBF) networks are considered for this purpose; the reason being the consistency of RBF networks in training. In other words, if an RBF network is trained with the same set of data several times, it will produce the same MSE for the test and training data. However, multi-layer perceptron (MLP) networks are highly dependent on their initial synaptic weights; thus, the evaluation of the results would be difficult. The adjustable parameters in RBF networks are: (1) maximum number of hidden neurons; (2) value of the spread constant (SC) which is commonly in the range of 0.01 to 5; and (3) a minimum value for the MSE, which is also referred to as the goal. The common value of 1 which is recommended for the SC is also considered in this article. By setting the goal value equal to 0 and the maximum number of hidden neurons equal to the number of available training vectors, the network will be trained until it reaches MSE=0 for the training data. In such a case, the MSE for the test data is used as a criterion to compare different subsets of control factors (x_i, i=1,..., k). Through this procedure, full models including all of the control factors and all of its subsets of k−1 element models, k−2 element models, etc. can be compared with each other. The model that yields the lowest MSE for the test data contains the significant factors. If the difference between the full model and a subset model is not

Table 1 Levels of the control and noise variables used in the robustly designed experiment

Variable	Low level	High level	Type
Parent population	20	50	Control
Parent/offspring ratio	1:1	1:7	Control
Selection type	Rank	Tournament	Control
Number of elites	2	6	Control
Crossover rate	0.5	0.85	Control
Mutation type	Uniform	Gaussian	Control
Number of factors	4	8	Noise
Number of responses	4	16	Noise
Constraint width (% of target)	5	15	Noise

Table 2 Final parameter settings for the genetic algorithm (GA)

Parameter	Value
Parent population	20
Parent/offspring ratio	1:7
Selection type	Tournament
Number of elites	2
Crossover rate	0.85
Mutation type	Gaussian

significant, the full model will be preferred. It should be noticed that, when a factor is eliminated from the full model, a replicated model in a reduced space is generated automatically. In this situation, one should be careful not to consider one replicate as the training and the other replicate as the test data set simultaneously. If this happens, then our judgment will be flawed.

3.3.2 Designing the most appropriate ANN to estimate each response

After identifying the significant factors for each response, the best network with the lowest MSE will be designed. At this step, one can consider either MLP or RBF networks to model the relationship between each response and its corresponding significant factors. To approve the appropriate training for each network, the network output for the test and training data should be plotted and compared against the desired data obtained from the initial experiment.

3.4 Combining different responses through desirability functions

One of the most powerful approaches in multiple-objective decision-making (MODM) is the desirability function, which transforms a multiple-objective problem into a single-objective problem. Actually, this approach consists of transforming each one of the m responses into its desirability according to a specific target. Individual desirability, $d_j(y_j)$, j=1, 2,..., m, transforms a response into a scaleless value in the range $0<d_j(y_j)<1$. A higher $d_j(y_j)$ value shows a more desirable response. Finally, the

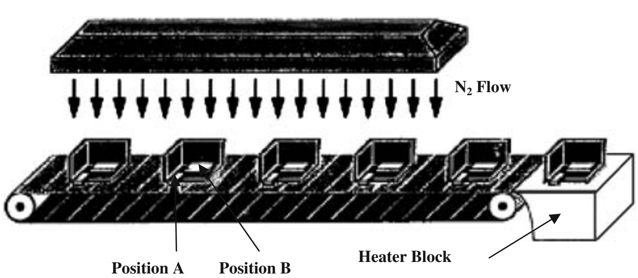

Fig. 3 Wire bond heating system [4]

Table 3 Factors and levels for the Box-Behnken experimental design

Factor	Name	Units	Low level	High level
A	Flow rate	SCFM	40.0	120.0
B	Flow temp.	°C	200.0	450.0
C	Block temp.	°C	150.0	350.0

individual desirabilities are combined via the use of additive or multiplicative or mixed models that produces a single-value total desirability $D(x)$.

3.4.1 Defining individual desirability functions for quantitative responses

For quantitative responses, the use of individual desirability functions proposed initially by Derringer and Suich [5] and modified later by Ortiz et al. [17] is recommended. Derringer and Suich [5] define the individual desirability as:

$$d_j(\widehat{y}_j(x)) = \begin{cases} 0 & \text{if } \widehat{y}_j(x) \le y_{\min j} \\ \left(\dfrac{\widehat{y}_j - y_{\min j}}{y_{\max j} - \widehat{y}_j}\right)^r & \text{if } y_{\min j} \le \widehat{y}_j(x) \le y_{\max j} \\ 1 & \text{if } \widehat{y}_j(x) \ge y_{\max j} \end{cases}$$

for the one-sided case, and:

$$d_j(\widehat{y}_j(x)) = \begin{cases} \left(\dfrac{\widehat{y}_j - y_{\min j}}{T_j - y_{\min j}}\right)^s & \text{if } y_{\min j} \le \widehat{y}_j(x) \le T_j \\ \left(\dfrac{\widehat{y}_j - y_{\max j}}{T_j - y_{\max j}}\right)^t & \text{if } T_j \le \widehat{y}_j(x) \le y_{\max j} \\ 0 & \text{otherwise} \end{cases}$$

for the two-sided case, where $\widehat{y}_j$, j=1,..., m is the output of the jth network and $y_{\min j}$, $y_{\max j}$, and T_j are the minimum,

Table 4 Experimental runs

Flow rate	Flow temp.	Block temp.	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
40	200	250	139	103	110	110	113	126
120	200	250	140	125	126	117	114	131
40	450	250	184	151	133	147	140	147
120	450	250	210	176	169	199	169	171
40	325	150	182	130	122	134	118	115
120	325	150	170	130	122	134	118	115
40	325	350	175	151	153	143	146	164
120	325	350	180	152	154	152	150	171
80	200	150	132	108	103	111	101	101
80	450	150	206	143	138	176	141	135
80	200	350	183	141	157	131	139	160
80	450	350	181	180	184	192	175	190
80	325	250	172	135	133	155	138	145
80	325	250	190	149	145	161	141	149
80	325	250	180	141	139	158	140	148

Table 5 Mean square error (MSE) for the test data

Factors	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
A, B, C	446.0	460.3	419.0	143.1	67.4	21.7
B, C	5.6	15.8	139.7	1,135.4	206.6	151.2
A, C	4,049.4	1,551.0	415.7	3,117.5	1,379.1	776.5
A, B	260.9	3,501.5	1,141.0	284.8	215.1	1,169.3

maximum, and target values for the *j*th response, respectively. In the above equation, *r*, *s*, and *t* indicate the weights that allow for linear (s=t=1) or nonlinear behavior between a bound ($y_{\min j}$ or $y_{\max j}$) and the target (T_j). Ortiz et al. [17] added a penalty term to the model proposed by Derringer and Suich [5] which helps the GA to maintain an infeasible solution while not allowing it to have a total desirability higher than a feasible solution. The penalty term recommended by Ortiz et al. [17] is as follows:

$$p_j(\widehat{y}_j) = \begin{cases} c + \left|\dfrac{\widehat{y}_j - y_{\min j}}{T_j - y_{\min j}}\right| & -\infty \le \widehat{y}_j \le y_{\min j} \\ c, & y_{\min j} \le \widehat{y}_j \le y_{\max j} \\ c + \left|\dfrac{\widehat{y}_j - y_{\max j}}{T_j - y_{\max j}}\right| & y_{\max j} \le \widehat{y}_j \le +\infty \end{cases}$$

where c is arelatively small constant, such as 0.0001, which forces $p_j(\widehat{y}_j)$ to be greater than 0.

3.4.2 Defining individual desirability functions for qualitative responses

As we see, the outputs of neural networks for qualitative responses are in the form of an n+1 element vector. Therefore, at this point, defining the desirability of a vector is needed. The individual desirability for a vector can be computed as following:

$$d_j(\bar{y}_j) = \{1 - \frac{\left[(\bar{y}_j - \overline{T}_j)^{\mathrm{T}}(\bar{y}_j - \overline{T}_j)\right]^{1/2}}{\left[(\overline{T}_j - \overline{W}_j)^{\mathrm{T}}(\overline{T}_j - \overline{W}_j)\right]^{1/2}}\}$$

where $\bar{y}_j$ is the output vector of the *j*th network, $\overline{T}_j$ is the target vector, and $\overline{W}_j$ is the complement of the target vector. This formula first calculates the normalized statistical distance of the output vector $\bar{y}_j$ from its target vector $\overline{T}_j$; next, by subtracting it from 1, it determines its individual desirability. The target vector $\overline{T}_j$ is a vector with an element

Table 6 Significant control factors for each response

Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
B, C	B, C	B, C	A, B, C	A, B, C	A, B, C

Table 7 Properties of the six final neural networks

Network	Output	Type	No. of neurons in the hidden layers	MSE	
				Test	Training
1	Y_1	MLP	3, 6	0.55	0.08
2	Y_2	MLP	5, 6	0.55	0.11
3	Y_3	MLP	5, 2	1.98	0.00
4	Y_4	MLP	4, 6	1.60	0.20
5	Y_5	MLP	7, 6	14.00	0.03
6	Y_6	MLP	7, 4	1.14	0.28

equal to 1 and n elements equal to 0, where the location of the 1 in the vector should be defined by the expert according to the most desirable fuzzy set. $\overline{W}_j$ is the complement of the target vector with n elements equal to 1 and one element equal to 0, where the location of the 0 element is the same as the location of the 1 element in the target vector. The penalty term is set equal to c for all qualitative responses.

After determining the individual desirabilities for qualitative and quantitative responses and considering the penalty terms, the total desirability defined by $D^*(x)$ can be computed as follows:

$$D_{\mathrm{DS}}(x) = [d_1(Y_1(x))d_2(Y_2(x))\ldots d_m(Y_m(x))]^{1/m}$$

$$P(x) = \left[(p_1(y_1)p_2(y_2)\ldots p_m(y_m))^{1/m} - c\right]^2$$

$$D^*(x) = D_{\mathrm{DS}}(x) - P(x),$$

where $D_{\mathrm{DS}}(x)$, the overall desirability defined by Derringer and Suich [5], is the geometric mean of the individual desirabilities ($d_j(y_j)$). Furthermore, $P(x)$, which is the combined function of the individual fitted responses, is the overall penalty function. The function $P(x)$ shows the overall severity of infeasibility. The individual penalties $p_j(\hat{y}_j)$ ensure a nonzero overall penalty $P(x)$. On the other hand, $P(x)$ will be zero for any feasible solution. The total desirability $D^*(x)$ is the criterion for comparing different solutions and is the function which we want to maximize.

3.5 Optimizing via GA

A GA is chosen to perform the optimization for two main reasons. First of all, gradient-based optimization methods,

Table 8 Computed MSE from the regression models

Response	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
MSE	78.3120	38.6450	63.3360	33.2550	8.3890	7.5822

Table 9 Minimum, maximum, and target values for the six responses, along with their associated desirability values

Response	$y_{\min j}$	T_j	$y_{\max j}$	$d_j(y_{\min j})$	$D_j(T_j)$	$d_j(y_{\max j})$
Y_1	185	190	195	0.0	1.0	0.0
Y_2	170	185	195	0.0	1.0	0.0
Y_3	170	185	195	0.0	1.0	0.0
Y_4	185	190	195	0.0	1.0	0.0
Y_5	170	185	195	0.0	1.0	0.0
Y_6	170	185	195	0.0	1.0	0.0

such as GRG, cannot be used because they require response surfaces to compute the gradient and direction of improvement. However, when neural networks are used, there would be no response surface for this reason. Secondly, GAs are known as a potent heuristic search method for optimizing highly nonlinear and complex functions.

A GA has various parameters whose values need to be determined before the optimization phase begins. Different authors, including Ortiz et al. [17], have proposed the use of a robustly designed experiment to determine the best settings for a GA's parameters. Therefore, we have incorporated a robustly designed experiment to find the best settings for the parameters. The GA's control and noise variables and their levels are shown in Table 1.

The GA performance measures are the same as in Ortiz et al. [17]. For more details, the reader can refer to that paper. The final parameter settings for the robust GA are shown in Table 2.

Finally, the tuned GA is run for 1,000 repetitions and the most desirable solution with the highest total desirability would be the final solution. (All of the calculations and coding are performed in the MATLAB environment.)

4 Examples

The proposed approach is illustrated with two examples. The first example is from the literature, which includes only quantitative responses. Since we could not find any multiple-response problems with qualitative responses in the literature, a melt spinning process was simulated in the second example.

4.1 Optimization of a multiple-response semiconductor-manufacturing process

This example, which was discussed by Del Castillo et al. [4], is based around the wire-bonding process in the semiconductor industry. During this process, the manufacturer must assemble a hybrid module in a pre-molded package by bonding wires between the leads (position A, Fig. 3) and the silicon chips (position B, Fig. 3).

The control factors that influence the temperature at the wire bond are the N_2 flow rate (x_1), the N_2 temperature (x_2), and the heater block temperature (x_3). The responses for the experiment are Y_1=maximum temperature at position A, Y_2=beginning bond temperature at position A, Y_3=finish bond temperature at position A, Y_4=maximum temperature at position B, Y_5=beginning bond temperature at position B, Y_6=finish bond temperature at position B. To investigate the effect of the three control factors, a Box-Behnken design was used [1]. The control factors, along with their levels used in the design, is shown in Table 3. The experimental results are shown in Table 4.

Since there is no qualitative response, the preprocessing phase is skipped. In this design, three replicates are available at the center point. Hence, one of the runs is chosen randomly. To identify the significant control factors, the RBF networks were first designed. The MSE for the test data corresponding to the full model and models including two control factors are shown in Table 5.

Significant control factors for each of the six responses are shown in Table 6. Whenever there is no significant difference between a full model and a model including a subset of control factors, the full model is selected.

By comparing the significant control factors achieved by our approach and those which were used in the regression models of Del Castillo et al. [4], it can be seen that, except for Y_2 and Y_3, all of the models share the same factors. For these two responses, their model includes factor A, which is excluded by our approach. However, it should be noted that the regression multiplier calculated for factor A in both of their models is smaller than the rest of the multipliers.

Next, considering the control factors mentioned in Table 6 for each response, different MLP and RBF networks with different parameters are trained. The most

Table 10 Comparison of the final solutions

Approach	x	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	$D(x)$
Proposed	(68.97, 370.00, 286.06)	192.1	184.1	181.0	193.7	180.3	171.2	0.4168
Ortiz et al. [17]	(74.55, 472.90, 332.75)	187.0	176.7	173.8	192.9	174.2	186.2	0.4081
Del Castillo et al. [4]	(84.16, 450.00, 329.87)	186.0	174.5	172.0	192.6	173.0	185.0	0.3061

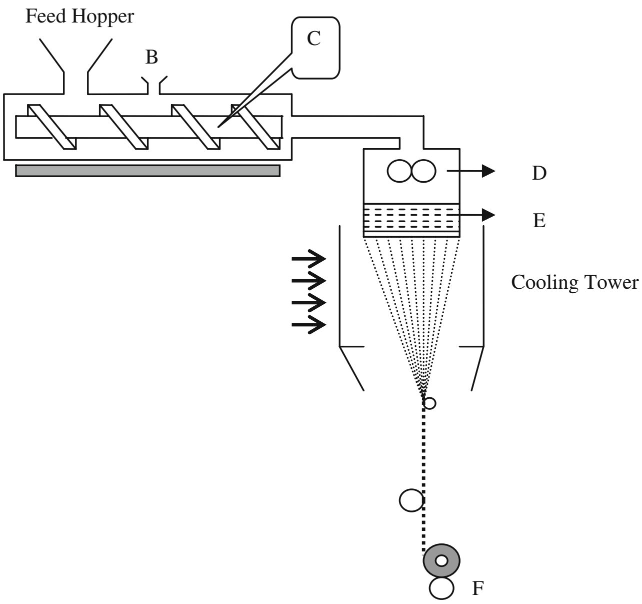

Fig. 4 Nylon 6 yarn manufacturing process

appropriate networks with the lowest MSE are as presented in Table 7.

As can be seen, MLP is selected as the best network for all of the responses. All of the MLP networks have two hidden layers with hyperbolic tangent activation functions. The output neurons have linear activation functions. The training algorithm in all of the six networks is Levenberg-Marquardt. The proportion of test data to the total data for the first three networks is 11.11% and for the next three networks is 15.39%.

The MSE of the six regression models (presented in [4]) are also computed and presented in Table 8. As can be seen, the computed MSE from the regression models are high, and this shows a poor fitness of the models. However, the six neural networks produce absolutely lower MSE, hence, they can approximate the process function more accurately.

To use the desirability approach, the process engineer selects the lower, upper, and target values for individual desirabilities, as shown in Table 9. Individual desirabilities are supposedly linear, with s=t=1 and c equal to 0.0001.

Table 10 compares the solution found using the revised desirability approach of Del Castillo et al. [4], the unconstrained desirability approach of Ortiz et al. [17], and our proposed approach.

As can be seen in Table 10, the total desirability achieved by the proposed approach shows its ability in optimizing multiple-response problems in comparison with two other leading approaches. The final solutions provided by Ortiz et al.'s and Del Castillo et al.'s approaches are close to each other. This is due to the fact that they share the same regression models for the six responses.

Table 11 Control factors and their levels in the experimental design

Factor	Name	Units	Low level	High level
A	Heater temp.	C°	250	290
B	Anti-static	%	6	8
C	Extruder speed	rev/min	400	500
D	Metering pump's rate	cm^3/min	700	900
E	Dimension of spinneret holes	µm	1,000	1,200
F	Take-up roll speed	rev/min	100	120

4.2 Optimization of a multiple-response nylon 6 yarn manufacturing process

The nylon 6 manufacturing process is a branch of melt spinning called chip spinning or extrusion. As illustrated in Fig. 4, the process begins with feeding the granules of nylon 6 polymer to the feed hopper. The nylon 6 polymer is then melted by being subjected to a heater (A) and is then mixed thoroughly by the rotation of a screw extruder (C). Next, the anti-static solution (B) is added to the mixture. Rotation of the screw extruder pushes the mixture forward in the manifold. The flow of molten polymer is aided by the metering pump (D). The molten polymer is then spun by passing through the spinneret (E) and takes the shape of the filament. The filaments next solidify by passing through the cooling tower. Finally, they are lubricated and taken up via the take-up roll (F).

The responses of the experiment which should be optimized are the yarn number, defined as weight per unit length and measured in dtex (Y_1), specific tension of the

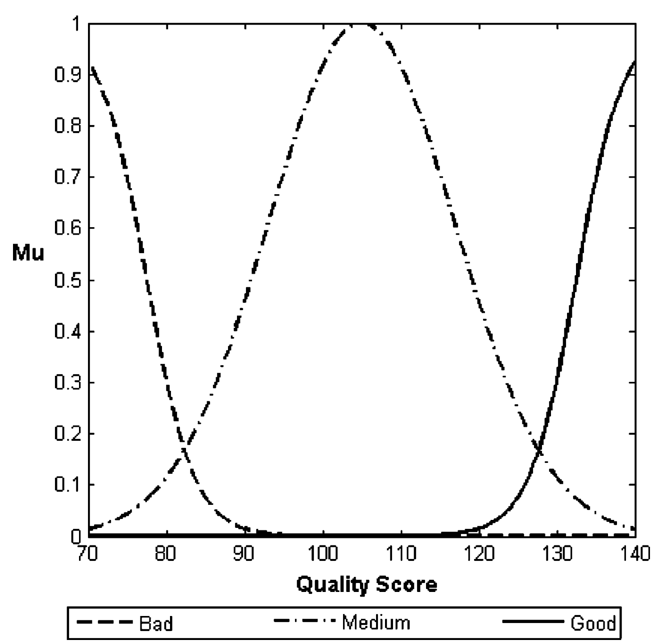

Fig. 5 Three membership functions of the fuzzy response (Y_3)

Table 12 Relevant vectors for the three fuzzy sets

Bad	Medium	Good
0.92414	0.01422	0.00000
0.84113	0.02554	0.00000
0.69706	0.04394	0.00000
0.50000	0.07238	0.00000
0.30294	0.11416	0.00000
0.15887	0.17242	0.00000
0.07586	0.24935	0.00000
0.03445	0.34529	0.00000
0.01527	0.45783	0.00000
0.00669	0.58127	0.00000
0.00292	0.70665	0.00000
0.00127	0.82258	0.00001
0.00055	0.91686	0.00002
0.00024	0.97853	0.00005
0.00010	1.00000	0.00010
0.00005	0.97853	0.00024
0.00002	0.91686	0.00055
0.00001	0.82258	0.00127
0.00000	0.70665	0.00292
0.00000	0.58127	0.00669
0.00000	0.45783	0.01527
0.00000	0.34529	0.03445
0.00000	0.24935	0.07586
0.00000	0.17242	0.15887
0.00000	0.11416	0.30294
0.00000	0.07238	0.50000
0.00000	0.04394	0.69706
0.00000	0.02554	0.84113
0.00000	0.01422	0.92414

yarn in CN/tex (Y_2), and yarn appearance condition (Y_3). The first two responses are quantitative and the third one, which shows the number of snarls in a unit of length and is defined by an expert observing the yarn under a microscope, is qualitative. The purpose of this experiment is to achieve a target value equal to 140 for Y_1, maximize Y_2, and achieve the best yarn condition with the least possible snarls (Y_3). The process engineer identifies six control factors that affect these three responses. The control factors are presented in Table 11.

Table 13 MSE for the test data

Model including factors	Y_1	Y_2	Y_3
A, B, C, D, E, F	3.26	4.12	0.02
B, C, D, E, F	5.75	40.35	0.21
A, C, D, E, F	3.50	2.53	0.06
A, B, D, E, F	4.35	1.16	0.09
A, B, C, E, F	1,815.90	15.66	0.23
A, B, C, D, F	540.45	1.33	0.03
A, B, C, D, E	41.37	3.01	0.03

Table 14 MSE of the test data for the three networks with identified significant factors

Response	Significant factors	MSE
Y_1	D, E, F	2.78
Y_2	A, D	4.01
Y_3	A, B, C, D	0.02

The equations considered for simulating the process and generating the data are as follows:

$$\begin{aligned}
Y_1 &= 81.66 + 28.33X_4 - 21.66X_6 + 95X_4X_5 + \varepsilon \\
Y_2 &= 35.55 + 33.63X_1 - 9.49X_3 - 7.86X_4 + 0.9X_5 \\
&\quad - 10.58X_1X_3 - 17.68X_2^2 + \varepsilon \\
Y_3 &= 97.56 + 25.56X_1 - 9.86X_2 + 42.88X_3 - 37.69X_4 + \varepsilon
\end{aligned}$$

where $\varepsilon \sim N(0, \sigma^2)$ and X_1, X_2,..., X_6 are defined as:

$$X_1 = \frac{A-220}{100} \qquad X_2 = \frac{B-4}{6} \qquad X_3 = \frac{C-300}{300}$$
$$X_4 = \frac{D-600}{400} \qquad X_5 = \frac{E-900}{400} \qquad X_6 = \frac{F-80}{60}$$

In the next step, a central composite design with a center point was chosen and the responses were computed using simulation equations. To generate qualitative values in the form of lingual expressions for Y_3, the interval 80 to 136, which contains the results of the third equation, is divided into three sections and their corresponding lingual expressions are used as follows:

$$\begin{aligned}
&Y_3 < 90 && \Rightarrow \text{bad} \\
&90 < Y_3 < 120 && \Rightarrow \text{medium} \\
&Y_3 > 120 && \Rightarrow \text{good}
\end{aligned}$$

The experimental results which contain 65 runs are not included in the paper; however, they are available upon reader request. Next, the qualitative response Y_3 is fuzzified. Three fuzzy sets and their membership functions

Table 15 Properties of the three trained networks

Network	No. of hidden layers	No. of neurons in each layer	Output layer's activation function	MSE	
				Test	Training
1	2	4, 3	Linear	0.95	0.48
2	2	5, 5	Linear	0.89	0.70
3	1	14	Sigmoid	0.00	0.00

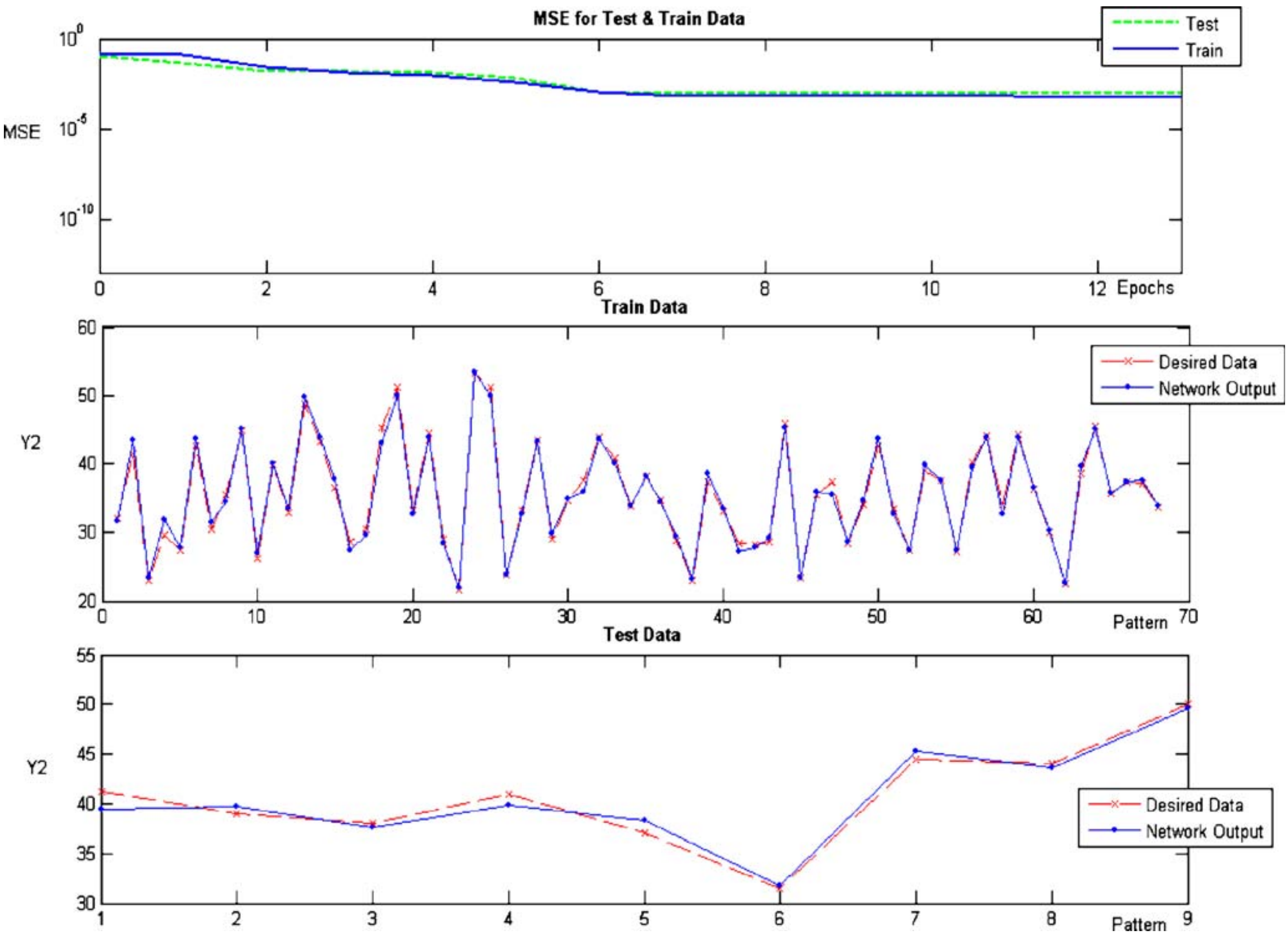

Fig. 6 Confirmative graphs for the second network

are defined below. Figure 5 illustrates the membership functions:

Y_3 Yarn appearance condition
$T(x)$ {good, medium, bad}
U [70, 140]

$$\mu_{\mathrm{bad}}(Y_3) = \frac{1}{1+\exp(1/3(Y_3-77.5))}$$

$$\mu_{\mathrm{medium}}(Y_3) = \exp\left(-\frac{(Y_3-105)^2}{2(12)^2}\right)$$

$$\mu_{\mathrm{good}}(Y_3) = \frac{1}{1+\exp(-1/3(Y_3-132.5))}$$

Subsequently, in order to feed these membership functions to the neural networks, they should be expressed as a vector. To do so, the interval [70, 140] is divided to 28 equal sections (n=28). According to Sect. 3.2, the 29 element vectors of the three fuzzy sets are as shown in Table 12.

To identify significant control factors, the MSE of the test data for the full model and models including five control factors (k−1) are given in Table 13.

According to Table 13, factors D, E, and F for response Y_1, A and D for response Y_2, and A, B, C, and D for response Y_3 are identified as significant factors. Next, the RBF networks were formed with significant factors as the inputs of each network. The results are shown in Table 14.

Table 16 Minimum, maximum, and target values for three responses with their associated desirability values

Response	Target	$y_{\min j}$	T_j	$y_{\max j}$	$d_j(y_{\min j})$	$D_j(T_j)$	$d_j(y_{\max j})$
Y_1	Target	130	140	150	0.0	1.0	0.0
Y_2	Max	40	50	-	0.0	1.0	1.0

Table 17 Final solution found by the GA

x^*	Approach	Simulation	$D_{\text{Approach}}(x)$
	(Y_1, Y_2, Y_3)	(Y_1, Y_2, Y_3)	
(368.13, 9.94, 398.21, 803.63, 1,297.50, 83.43)	(140.08, 61.27, Vector)	(142.90, 56.68, Vector)	0.8423

Since the MSE of the models with significant factors are not significantly lower than the MSE of the full model, the full models are chosen for the optimization. Subsequently, the most appropriate networks with the lowest MSE were trained. The best networks for all three responses are MLP with the Levenberg-Marquardt training algorithm. The activation functions of the hidden layer's neurons in all of the networks are hyperbolic tangent. Other properties of the networks are as shown in Table 15. The percentage of the test data to the total data in all three networks is 11.69%.

The appropriate training of the networks to avoid the memorization phenomenon has been controlled by plotting each network's output for the test and training input data and mapping them along their desired values. Furthermore, the MSE decrease trend has been controlled in order to not have an instant decrease, which accounts for memorization. The confirmative graphs for appropriate training of the second network are shown in Fig. 6.

The confirmative graphs for the rest of the networks of this example and also all six networks of the first example have not been included in this article to reduce its size, although they all account for the appropriate training.

The process engineer selects the lower, upper, and target values for individual desirabilities, as shown in Table 16.

For the qualitative response, the target vector T_3 and its complement W_3 are as follows. Individual desirabilities are assumed to be linear with s=t=1:

$$\overline{T}_3 = (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1)^{\mathrm{T}}$$

$$\overline{W}_3 = (1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 0)^{\mathrm{T}}$$

With the achieved parameter settings, the GA was run 1,000 times. The most desirable solution found by the algorithm, its predicted response values (defined by neural networks), and the actual response values (defined by simulation equations) are shown in Table 17. The output of the third network for the final solution is shown in Fig. 7.

As can be seen in Table 17, the predicted values of the first and second response achieved by the first and second networks are close to their actual values attained by simulation equations. In addition, the output of the third network for the final solution is close to the target vector. This shows the proper training of all three networks and approves the proposed approach in dealing with problems which include qualitative responses.

5 Conclusion

This paper proposes an approach for the optimization of multiple-response problems with or without qualitative responses. The approach considers a neural network for each response to estimate its relation with control factors, unconstrained desirability functions to combine different responses into one, and a genetic algorithm (GA) to perform the optimization. The proposed approach is novel because of three main aspects. First, it utilizes neural networks to estimate the responses of interest. Second, it identifies the significant control factors of each of the responses through the usage of radial bias function (RBF) networks. Finally, it can deal with problems which contain qualitative responses.

The performance of the proposed approach was evaluated through two manufacturing examples. The first example from the semiconductor manufacturing industry was chosen due to the high mean square error (MSE) of the regression models suggested by Del Castillo et al. [4], which shows the poor quality of description. As it was shown, the usage of neural networks results in a lower MSE; hence, its final solution would be more reliable.

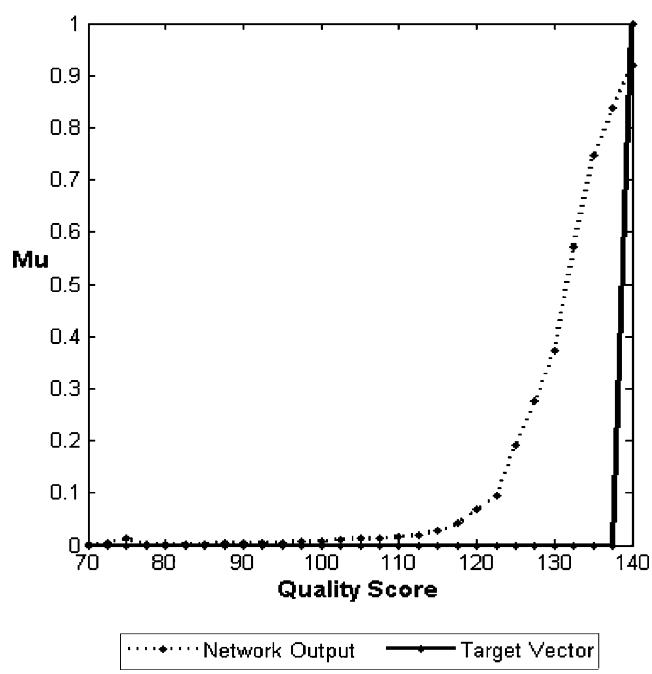

Fig. 7 Output of the third network for the final solution and the target vector

Furthermore, the performance of RBF networks in identifying the significant control factors of each response was compared with those of Del Castillo et al. [4]. In the second example from the synthetic thread manufacturing industry, an optimization problem was considered in the presence of a qualitative response. The approach was able to perform greatly and achieves the optimization goals. The reason for this lies in the closeness of the predicted response values defined by neural networks to their calculated values from simulation equations. Furthermore, the output of the third network in the form of a fuzzy set is much closer to its desired shape.

Notably, the proposed optimization approach only involves the location effect of the responses. Therefore, one can extend the approach to include the dispersion effect of the responses. In these situations, replications are needed to estimate the standard deviation of different responses at different settings of the control factors.

Acknowledgements Dr. Noorossana's research is partially supported by a grant from Iran National Science Foundations.

References

1. Box GEP, Behnken DW (1960) Some new three-level designs for the study of quantitative variables. Technometrics 2:455–475
2. Chiao CH, Hamada MS (2001) Analyzing experiments with correlated multiple responses. J Qual Technol 33(4):451–465
3. Del Castillo E, Montgomery DC (1993) A nonlinear programming solution to the dual response problem. J Qual Technol 25:199–204
4. Del Castillo E, Montgomery DC, McCarville DR (1996) Modified desirability functions for multiple response optimization. J Qual Technol 28(3):337–345
5. Derringer G, Suich R (1980) Simultaneous optimization of several response variables. J Qual Technol 12:214–219
6. Harrington EC Jr (1965) The desirability function. Ind Qual Control 21:494–498
7. Haykin S (1994) Neural networks. Macmillan College Publishing Company, New York
8. Khuri AI, Conlon M (1981) Simultaneous optimization of multiple responses represented by polynomial regression functions. Technometrics 23:363–375
9. Kim KJ, Lin DKJ (2006) Optimization of multiple responses considering both location and dispersion effects. Eur J Oper Res 169:133–145
10. Kim KJ, Byun JH, Min D, Jeong IJ (2001) Multiresponse surface optimization: concept, methods, and future directions. Tutorial, Korea Society for Quality Management
11. Leondes CT (1998) Fuzzy logic and expert systems applications. Academic Press, New York
12. Montgomery DC (1991) Design and analysis of experiments, 3rd edn. Wiley, New York
13. Montgomery DC (2005) Introduction to statistical quality control, 5th edn. Wiley, New York
14. Montgomery DC, Peck EA, Vinning G (2001) Introduction to linear regression analysis, 3rd edn. Wiley, New York
15. Myers RH, Montgomery DC (2002) Response surface methodology: process and product optimization using designed experiments, 2nd edn. Wiley, New York
16. Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7:308–313
17. Ortiz F, Simpson JR, Pignatiello JJ, Heredia-Langner A (2004) A genetic algorithm approach to multiple-response optimization. J Qual Technol 36:432–450
18. Pignatiello JJ Jr (1993) Strategies for robust multiresponse quality engineering. IIE Trans 25:5–15
19. Shah HK, Montgomery DC, Carlyle WM (2004) Response surface modeling and optimization in multiresponse experiments using seemingly unrelated regressions. Qual Eng 16 (3):387–397
20. Terano T, Asai K, Sugeno M (1991) Fuzzy systems theory and its application (translated by Charles G. Aschmann). Academic Press, New York, ISBN 0-12-685245-6
21. Tong LI, Hsieh KL (2000) A novel means of applying neural networks to optimize the multiresponse problem. Qual Eng 13 (1):11–18
22. Vining GG (1998) A compromise approach to multiresponse optimization. J Qual Technol 30(4):309–313