SCIENTIA
IRANICA

Optimizing Multi-Response Statistical Problems Using a Genetic Algorithm

S.H.R. Pasandideh[1] and S.T. Akhavan Niaki[*]

In this paper, two methods to solve multi-response statistical problems are presented. In these methods, desirability function, genetic algorithm and simulation methodology are applied. The desirability function is responsible for modeling the multi-response statistical problem, the genetic algorithm tries to optimize the model and, finally, the simulation approach generates the required input data from a simulated system. The methods differ from each other in controlling the randomness of the problem. In the first method, replications control this randomness, while, in the second method, the randomness is controlled by a statistical test. Furthermore, these methods are compared by designed experiments and the results are reported.

INTRODUCTION AND LITERATURE REVIEW

A usual problem in the real world environment involves selecting a set of input conditions (the x's being independent variables) which will result in a product with a desirable set of outputs (the y's being response variables). Essentially, this becomes a problem in the simultaneous optimization of the response variables, each of which depends upon a set of independent variables, $x_1, \cdots, x_p$. In this problem, the levels of the independent variables are to be selected, such that all the response variables optimize. In this case, however, the selected levels of the x's that optimize, for example, y_1, might not even come close to optimizing y_2.

As an example, in quality control environments, the goal may be to find the levels of the input variables (quality characteristics) of the process so that the quality of the product or responses has the desired characteristics. Also, in Response Surface Methodology (RSM) [1] the levels of the input variables are adjusted until the set of outputs are optimized. In most RSM problems, the form of relationship between the response and the independent variables is unknown. Thus, the first step in RSM is to find a suitable approximation. Usually, a low-order polynomial is employed. If there is curvature in the system, then, a polynomial of higher degree, mainly second order, is used. Then, by a sequential procedure and the method of steepest ascent or steepest descent, the best set of input for response is determined. RSM usually works well when one response is considered. Also, usually, RSM is not applied in complex cases, such as non-polynomial and higher-order or multi-modal functions.

While many real world problems involve the analysis of more than one response variable, most of the mathematical programming applications in the literature have focused on single response problems and few attempts have been made to solve multiple-objective statistical problems. These attempts can be classified into four categories [2,3].

The usual practice in the first category is to simplify the problem, selecting the most important response and ignoring the other responses or considering them as the model constraints. For example, one can refer to Hartmann and Beaumont [4] and Biles [5]. While Hartmann and Beaumont modeled the problem using a linear programming approach, Biles used this approach once in conjunction with a version of Box's complex method [6] and, alternatively, along with a variation of the gradient method. The proposed procedures of this category would generally lead to unrealistic solutions, especially when conflicting objectives are present. For example, in a capital investment problem with two objectives, profit maximization and risk minimization, it usually happens that the higher

---

1. *Department of Industrial Engineering, Sharif University of Technology, P.O. Box 11365-9414, Tehran, I.R. Iran.*
*. *Corresponding Author, Department of Industrial Engineering, Sharif University of Technology, P.O. Box 11365-9414, Tehran, I.R. Iran.*

the profit, the bigger the risk. For this reason, treating this problem using a single objective will lead to a poor solution.

The second category, where, basically, the objectives are aggregated into a single objective function, has been attempted several times in the literature, each time with relative success [3]. One of these attempts is called the weighted sum method and consists of adding all the objectives together using different weighting coefficients. Also, a variation of the goal programming method falls into this category. For instance, Clayton et al. [7], Rees et al. [8] and Baesler and Sepulveda [9] used this approach, along with other optimization methods. Baesler and Sepulveda integrated goal programming and Genetic Algorithm (GA) methods to solve the problem. Moreover, they used some statistical tests to control the random nature of the problem. Another method in this category is the goal attainment method, in which, in addition to the goal vector for each response, a vector of weights, relating the relative-under or relative-over attainment of the desired goals, must be elicited from the decision maker. The most serious pitfall of the method in this category is the importance of the responses and, hence, the determination of the weights in the objective function.

In the third category, some multi-attribute value functions are used. Mollaghasemi et al. [10] used a multi-attribute value function representing the decision-maker preferences. Then, a gradient search technique was applied to find the optimum value of the assessed function. Moreover, Mollaghasemi and Evans [11] proposed a modification of the multi-criteria mathematical programming technique, called the STEP method, which works by interaction with the decision-maker. Teleb and Azadivar [12] proposed an algorithm based on the constrained scalar simplex search method. This method works by calculating the objective function value in a set of vertices of a complex. It moves towards the optimum by eliminating the worst solution and replacing it with a new and better solution. The process repeats until a convergence criterion is met. Boyle [13] presented a method called the Pair-wise Comparison Stochastic Cutting Plane (PCSCP), which combines features from interactive multi-objective mathematical programming and response surface methodology.

In the fourth category, a search-heuristic algorithm is basically used. Cheng et al. [14] presented a neuro-fuzzy and GA method for optimizing the multiple response problems. Schaffer [15] introduced a new method, called the Vector Evaluated Genetic Algorithm (VEGA), which differed from the simple GA method by way of chromosome selection. Allenson [16] used a population-based modeling on VEGA, in which gender was used to distinguish between the two objectives of a problem, consisting of planning a route composed of a number of straight pipeline segments. In this method, only male-female mating is allowed and gender is randomly assigned at birth. Fourman [17] suggested use of a GA-based method on the lexicographic ordering problem. In his approach, the designer ranks the objectives in order of importance. The optimum solution is then obtained by optimizing the objective function, starting with the most important and proceeding according to the assigned order of importance. Periaux et al. [18] proposed a GA-based method that uses the concept of game theory to solve a bi-objective optimization problem. Coello [19] proposed a min-max strategy with a GA. In this method, the decision maker has to provide a predefined set of weights that will be used to spawn several small subpopulations that evolve separately, each trying to converge to a single point. Fonseca and Fleming [20] proposed a GA scheme, in which the rank of an individual corresponds to the number of chromosomes in the current population by which it is dominated. Kim and Rhee [21] proposed a method based on the desirability function and GA and applied this method to optimize a welding process. Heredia-Langner et al. [22] presented a model-robust alphabetically-optimal design with a GA. This technique is useful in situations where computer-generated designs are most likely to be employed. In summary, the review of the literature in this category reveals that the Genetic Algorithm (GA) method has a specific role and works successfully.

In the following section, a brief description of the desirability function approach will be given. Then, a multi-response statistical optimization problem is modeled through the desirability function method. After that, two procedures will be presented, both based on the GA approach, to solve the problem. In order to evaluate the performance of the proposed procedures and to compare them, some numerical examples will be solved and the results will be reported. Finally, the conclusion and some recommendations for future research are reported.

DESIRABILITY FUNCTION

The desirability function approach is one of the most widely used methods in industry for dealing with the optimization of multiple-response problems. It is based on the idea that the quality of a product that has multiple quality characteristics is completely unacceptable if one of the characteristics lies outside the desired limits. This method assigns a score to a set of responses and chooses factor settings that maximize that score.

In order to describe the desirability function approach mathematically, suppose each of the k response

variables are related to p independent variables by the following equation:

$$y_{ij} = f_i(x_1, \cdots, x_p) + \varepsilon_{ij},$$

$$i = 1, \cdots, k, \qquad j = 1, \cdots, n_i, \tag{1}$$

where y_{ij} is the jth observation on the ith response and f_i denotes the relationship between the ith response, y_i and $x_1, \cdots, x_p$. The parameter, n_i, is the maximum number of observations for each of the k responses and ε_{ij} is an error term with mean $E(\varepsilon_{ij}) = 0$ and variance $\mathrm{VAR}(\varepsilon_{ij}) = \sigma_i^2$, such that one can relate the average response to the p independent variables by the following equation:

$$\eta_i = f_i(x_1, \cdots, x_p), \quad i = 1, \cdots, k. \tag{2}$$

A desirability function, $d_i(y_i)$, assigns numbers between 0 and 1 to the possible value of each response, y_i. The value of $d_i(y_i)$ increases as the desirability of the corresponding response increases. The overall desirability, D, is defined by the geometric mean of the individual desirability values shown in the following equation:

$$D = (d_1(y_1) \times d_2(y_2) \times \cdots \times d_k(y_k))^{\frac{1}{k}}, \tag{3}$$

where k denotes the number of the responses. Note that if a response, y_i, is completely undesirable, i.e., $d_i(y_i) = 0$, then, the overall desirability value is zero.

Depending on whether a particular response, y_i, is to be maximized, minimized or assigned a target value, different desirability functions can be used. Derringer and Suich [23] introduced a useful class of desirability functions.

There are two types of transformation from y_i to $d_i(y_i)$, namely; one-sided and two-sided transformation. The one-sided transformation is employed when y_i is to be maximized or minimized and the two-sided transformation is used when y_i is to be assigned a target value.

In a two-sided transformation, assume l_i and u_i to be the lower and upper limits and t_i be the target value of the response, y_i, such that $l_i < t_i < u_i$. Then, the desirability function is defined as the following equation:

$$d_i(y_i) = \begin{cases} 0, & y_i < l_i \\ \left(\frac{y_i - l_i}{t_i - l_i}\right)^s, & l_i \le y_i \le t_i \\ \left(\frac{y_i - u_i}{t_i - u_i}\right)^t, & t_i \le y_i \le u_i \\ 0, & y_i > u_i, \end{cases} \tag{4}$$

where the exponents, s and t, determine how strictly the target value is desired and that the user must specify their values. For $s = t = 1$, the desirability
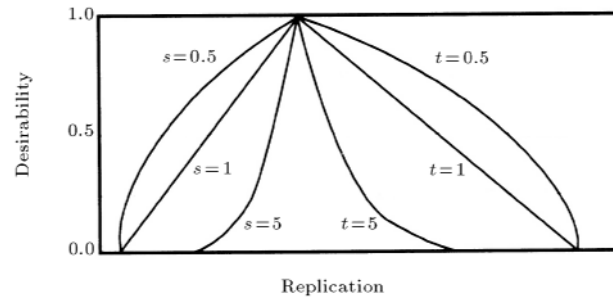
Figure 1. Graph of the two-sided transformation.

function increases linearly towards t_i. For $s < 1$ and $t < 1$, the function is convex and, for $s > 1$ and $t > 1$, the function is concave. This function for different values of s and t is graphed in Figure 1.

In Figure 1, the value of t_i is chosen to be in almost the lower 25% of the interval between l_i and u_i. From Figure 1, one can see that the large values of s and t are chosen for situations in which one needs to have the response near t_i. In situations in which one desires to have the response near l_i, the values of t and s are chosen small. Medium values are used for s and t (close to one) when a case between the above two extremes is desired. Furthermore, if the user wants to have the response approaches to t_i very quickly and any value for the response greater than t_i and less than u_i is desirable, then, the value of s is chosen to be large and the value of t to be small.

Similarly, one can define one-sided desirability functions in cases of minimizing or maximizing. It should be noted that while some modified versions of the desirability functions are useful for situations in which the exact mathematical methods of optimization are used, the introduced basic desirability functions are good enough for the search methods applied for optimization problems [24]. For a good reference, see [23].

PROBLEM MODELING

The candidate problem in the framework of this research has two main characteristics. First, the problem itself is a computer simulation of a real world problem. For example, a real world production system is simulated, in which events occur stochastically and different factors are affecting multiple responses simultaneously and where the goal is to determine the levels of the factors that optimize the responses. In this case, when different inputs are selected as the levels of the factors, the values of the responses are observed by simulation. Second, the levels of the factors can be modeled by a real variable. A real example of the problems having these characteristics is a quality control process, in which the reaction time, the temperature and the percent catalyst are the input variables and the responses are the percent conversion

and thermal activity of the process. In this situation, one may first want to simulate the production process and, then, determine the input variable levels, such that the outputs are maximized, minimized or set to a target value.

In the modeling phase of a multi-response statistical optimization problem, the objective function is transformed to a single function, using the desirability function framework. First, one must identify all the factors that make up the input of the problem. These factors are:

1. The independent variables, $x_1, \cdots, x_p$;

2. The lower and upper bounds of the independent variables ($L(x_h)$ and $U(x_h)$);

3. The output of the problem. This output is the response variables denoted by $y_1, \cdots, y_k$;

4. One-sided or two-sided desirability functions for each response. It is obvious that a one-sided or two-sided transformation for each response depends on the nature of the objective of the problem.

Then, the mathematical model of the problem becomes:

$$\max D = \sqrt[k]{(d_1(y_1) \times d_2(y_2) \times \cdots \times d_k(y_k))}$$

s.t.:

$$L(x_h) \leq x_h \leq U(x_h),$$

$$h = 1, 2, \cdots, p. \tag{5}$$

Based on the lowest error criterion, two methods are presented to solve the above model (Equation 5) in the next section.

METHODOLOGY

In order to solve the mathematical model of the problem (Equation 5), first, a simulation approach is employed to generate output response values for all the objectives in the problem. Note that the problem has a stochastic nature. This means that if the set of $x_1, \cdots, x_p$ is fixed, then, in each of the simulation execution cases, the set of $y_1, \cdots, y_k$ may be different.

Because of the stochastic nature of the model, one needs to apply a heuristic-search algorithm to solve it. Among these algorithms, the GA is shown to be successful in optimizing multi-response problems, especially when the objectives are combined into one objective [3].

The usual form of GA was described by Goldberg [25]. Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and natural genetics. GA is different from conventional search techniques in a sense that it starts with an initial set of random solutions called population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. To create the next generation, new chromosomes, called offsprings, are formed by either a crossover or mutation operator. A new generation is formed according to the fitness values of the chromosomes. After several generations, the algorithm converges to the best chromosome.

Knowing that a GA method needs scalar fitness information to work, the simplest idea would be to combine all the responses into a single one. In this methodology, the desirability approach is used for combining all the responses. Desirability functions have many advantages in comparison to other combining techniques. They have, in particular, a very flexible role. It means that one can, simultaneously, maximize some of the responses, minimize others and set target values to some of them.

Two GA methods are presented, which are different in structure, especially in controlling the stochastic nature of the problem. These methods have many similarities described in the following sub-sections. However, in the first method, the random nature of the problem is controlled by replication and, in the second method, the randomness is controlled by some multiple comparison statistical tests.

Initial Conditions

The initial information required to start the GA methods is as follows:

1. Population size: This is the number of the chromosomes or scenarios that are kept in each generation, denoted by N;

2. Number of replications: This is the number of the simulation replications of each scenario, denoted by n;

3. Crossover rate: This is the probability of performing a crossover in the GA method, which is denoted by P_c;

4. Mutation rate: This is the probability of performing mutation in the GA method, denoted by P_M.

Chromosome

In the GA methods, as a solution to the problem, one defines chromosomes or scenarios as being a set of the values for $x_1, \cdots, x_p$.

Initial Population

Generating an initial population of solutions or scenarios is the first stage towards starting the optimization process. The number of the scenarios is N and the scenarios are selected randomly to cover a wide range of solutions. Generally, users have a good idea where they can find good solutions. Some of the user's suggestions can be considered in the initial population.

In order to obtain the total desirability value of scenario j, $x_{1j}, x_{2j}, \cdots, x_{pj}$, after its preparation, it is simulated in n replications. Then, in each replication, the response variables are generated and the desirability value of each response is determined. To do this, the following parameters are defined:

x_{ij} is the input variable, i, in scenario j, $i = 1, \cdots, p$, $j = 1, \cdots, N$;

y_{ijr} is the response variable, i, in scenario j and in replication r, $i = 1, \cdots, k$, $j = 1, \cdots, N$, $r = 1, \cdots, n$;

d_{ijr} is the desirability, i, in scenario j and in replication r, $i = 1, \cdots, k$, $j = 1, \cdots, N$, $r = 1, \cdots, n$ and its value is obtained based on the desirability function characteristics described in the previous section;

D_{jr} is the total desirability value of scenario j in replication r, $j = 1, \cdots, N$, $r = 1, \cdots, n$;

$\overline{D}_j$ is the mean of the total desirability in scenario j, $j = 1, \cdots, N$ and it is calculated by $\overline{D}_j = \sum_{r=1}^{n} D_{jr}/n$;

S_j is the standard deviation of the total desirability in scenario j, $j = 1, \cdots, N$.

Crossover and Mutation

In a crossover process, it is necessary to mate pairs of chromosomes to create offspring. This is performed by selecting a pair of chromosomes from the generation randomly and with probability P_c. There are many different types of crossover operators. In this research, a variation of the crossover operators are used that work very well when the chromosome is represented using real and not binary code [26]. To define this operator precisely, assume that chromosomes A and B, presented, respectively, in the forms $[a_1, a_2, \cdots, a_p]$ and $[b_1, b_2, \cdots, b_p]$, are selected for the crossover operation. This crossover operator, using the following relations, will create the new chromosomes, C and D:

$$C = \lambda A + (1 - \lambda)B, \qquad (6)$$

$$D = (1 - \lambda)A + \lambda B, \qquad (7)$$

where λ is a parameter ranging between 0 and 1. One of the advantages of this crossover operator is that the new chromosomes, C and D, will be feasible.

Mutation is the second operation in the GA method for exploring new solutions. In mutation, a gene is replaced with a randomly selected number within the boundaries of the parameter [26]. More precisely, assume a specific gene, such as a_j, is selected for mutation, then, the value of a_j is changed to the new value, a_j^*, according to Equations 8 and 9, randomly and with the same probability:

$$a_j^* = a_j + (u_j - a_j) \times r \times \left(1 - \frac{i}{\max \text{gen}}\right), \qquad (8)$$

$$a_j^* = a_j - (a_j - l_j) \times r \times \left(1 - \frac{i}{\max \text{gen}}\right), \qquad (9)$$

where l_j and u_j are the lower and upper limits of the specified gene, r is a uniform random variable between 0 and 1, i is the number of the current generation and max gen is the maximum number of generations. Note that the value of a_j is transferred to its right or left randomly by Equations 8 and 9, respectively and r is this percentage. Furthermore, $1 - \frac{i}{\max \text{gen}}$ is an index with a value close to one in the first generation and close to zero in the last generation, which makes large mutations in the early generations and almost no mutations in the last generations.

Objective Function Evaluation

After producing the new chromosomes by crossover and mutation processes, they must be simulated. The value of response variables, desirability functions and total desirability for each of the new scenarios can be generated by simulation.

Chromosomes Selection

In the next phase of the methodology, the chromosomes are selected for the next generation. This selection is based on the fitness function value of each chromosome. The difference of the two proposed GA methods appears in this phase.

In the first method, the fitness function is considered, to be the total desirability, $\overline{D}_j$, of scenario j and, based on the better value of $\overline{D}_j$, N chromosomes are selected from the old and new ones deterministically. However, in the second method, because of the random nature of $\overline{D}_j$s the chromosomes are selected statistically. In this regard, first, the $\overline{D}_j$s are statistically compared and, then, the ones with better values are selected. To do this, a multiple-comparison statistical test [1] is applied to control the random nature of the situation and chromosomes are grouped, such that there is no statistical difference within the groups but there exist differences among different groups.

The multiple-comparison test used in the second method is Tukey's test. Tukey's test determines a

critical value, T_α, such that the probability of making a type I error is α for all comparisons. Any pairs of $\overline{D}_j$s that differ by less than T_α are statistically equal. The value of T_α is determined by the following equation.

$$T_\alpha = q_\alpha(N, f) \times \frac{\sqrt{\text{MSE}}}{\sqrt{n}}, \tag{10}$$

where $q_\alpha(N, f)$ is the upper percentage point of the Studentized Range Statistic [17], N is the population size, MSE is the mean squared error of $\overline{D}_j$s and f is the degrees of freedom associated with MSE, which is equal to $n(N - 1)$.

Now, based on the fitness function value, the chromosomes are ranked in ascending order and grouped with a difference between their fitness function values less than T_α, accordingly. In this way, groups of chromosomes are created that are not statistically different.

In order to generate chromosomes for the next generation, the roulette-wheel-selection technique [26,27], which is the most common selection technique, is applied. In this technique, first, the probability of selecting group k, p_k, is calculated by the following equation:

$$p_k = \frac{\sum_{\forall j \in \text{Group } k} \overline{D}_j}{\sum_{j=1}^{N} \overline{D}_j}. \tag{11}$$

Then, by the roulette-wheel-selection method, a group is chosen randomly and its best chromosome is selected, based on its fitness function value. This is done N times, making a generation with N chromosomes.

After the next generation is created, the crossover and mutation operators will operate on the new generation again and the selection phase will be repeated. This loop continues until the stopping criteria are met.

Stopping Criteria

The last step in the methodology is to check if the selected method has found a solution that is good enough to meet the user's expectations. Stopping criterion is a set of conditions such that when the method satisfies them, a good solution is obtained. In this research, one stops when no improvement in fitness function values of several consecutive generations is observed. The number of sequential generations depends on the specified problem and the expectations of the user.

PERFORMANCE OF THE PROPOSED METHODS

The performance of the proposed methods is studied in the following section.

Polynomial Examples

Since most of the applications of the new methods are in the quality control and RSM environments and considering the fact that polynomials are used in these situations, in order to evaluate the performance of the new methods and to compare them, different polynomial examples are applied. The examples of the experiments are polynomial in nature such that when the values of the independent variables are fixed, the expected values of the responses are known. For example, consider three responses (output variables) as polynomial functions of three independent variables (input variables) as follows:

$$y_1 = x_1^5 + x_1^4 + x_2^3 + x_3^6 + x_1 x_2 x_3 + \varepsilon_1,$$

$$y_2 = x_1^2 + x_1 x_3 + x_2 x_3 + x_1 x_2 x_3 + \varepsilon_2,$$

$$y_3 = x_1^3 + x_2^3 + x_3^4 + x_1 x_2^2 x_3^4 + \varepsilon_3,$$

where ε_1, ε_2 and ε_3 are the error terms with the following distributions; $\varepsilon_1 \sim N[0,1]$, $\varepsilon_2 \sim \text{EXP}(0.10)$, $\varepsilon_3 \sim N[2,4]$, and the input variables ranging in $0 \leq x_1 \leq 5$, $0 \leq x_2 \leq 3$ and $0 \leq x_3 \leq 4$. Note that for fixed values of the independent variables as $x_1 = 2$, $x_2 = 1$ and $x_3 = 3$, the average value of responses will be $T_1 = E(y_1) = 784$, $T_2 = E(y_2) = 19.1$ and $T_3 = E(y_3) = 254$. In this case, based on the fixed values of the response being $T_1 = 784$, $T_2 = 19.1$ and $T_3 = 254$ and applying a two-sided desirability function described in previous sections, the method that reaches closer to these values of the independent variables has a better performance than the other.

Thus, in order to compare the methods, a performance measure is defined as:

$$Q^2(m) = \sum_{j=1}^{p} (x_j(m) - x_j(a))^2, \tag{12}$$

where $x_j(a)$ is the given value and $x_j(m)$ is the obtained value of the input variable, j. Then, the method with a lower value of $Q^2(m)$ performs better.

In order to obtain results that are more reliable, the experimental examples are classified based upon the order of the polynomials and the degree of the variability of the example. Both factors are considered to have two levels, low and high. When the order of the polynomial is less than, or equal to, five, then, this factor is set to a low level, otherwise, it is set to its high level. The variability of the example is defined in terms of the sum of the error term's variances. Furthermore, the low and high levels of this factor are set when it is less than, or equal to, 50 and greater than 50, respectively. Therefore, different examples are considered in four classes. Class 1 contains examples in which both factors are set to their high levels. Class 2

consists of examples with both factors at their low levels. In Class 3, the order is high and the variability is low and, finally, Class 4 is designed to contain examples with the order at its low level and variability at its high level.

Then, both methods are applied to the examples of these classes several times, each time calculating the performance measure in Equation 12 and statistically comparing the two methods. In each class, five different polynomial functions are employed in four types of replications, making the total number of examples 80. As the case in all GA applications, in each example, the values of the mutation and crossover rate are chosen by trial and error.

For all of the examples, a two-sided desirability function is applied with the parameters s and t both being equal to one and the type I error in Tukey's test being equal to 0.05. Moreover, the value of the responses in each method is generated by random numbers from the error term's polynomials.

Tables 1 to 4 show the performance of each method when applied in different classes with different numbers of replication. Table 5 shows the results of the comprehensive case, where the two methods are applied to the problems of all classes in different replications. In addition, Figures 2 to 6 show the value of the performance measure vs. the number of replications being equal to 5, 10, 25 and 50 in different classes and in the comprehensive case. The figures are the graphical representation of the tables. For instance, consider Table 1 or Figure 2. Five examples in class one are run by the two methods. These examples are applied to a different number of replications. For each case, the error of the method is determined by Equation 12 and the corresponding sum of the error terms is shown in Figure 2. For example, 0.3714 is the sum of the error terms in five examples that have been run by the 1st method and the number of replications is equal to 5. Either Table 1 or Figure 2 shows that the 1st method is

Table 1. Performances in Class 1.

Methods	No. of Replications			
	5	10	25	50
First Method	0.3714	0.5495	0.4915	0.7555
Second Method	0.2318	0.3160	0.5773	1.2325

Table 2. Performances in Class 2.

Methods	No. of Replications			
	5	10	25	50
First Method	0.4363	0.5787	0.2095	0.1357
Second Method	2.0951	0.1875	0.1242	0.1707

Table 3. Performances in Class 3.

Methods	No. of Replications			
	5	10	25	50
First Method	0.0002	0.0003	0.0020	0.0039
Second Method	1.2048	0.4893	0.3762	2.9873

Table 4. Performances in Class 4.

Methods	No. of Replications			
	5	10	25	50
First Method	6.4144	0.4614	0.2316	0.1873
Second Method	4.1841	3.0127	0.1989	1.093

Table 5. Performances in the comprehensive case.

| Methods	No. of Replications			
	5	10	25	50
First Method	7.2223	1.5899	0.9346	1.1824
Second Method	7.7158	4.0055	1.2766	5.4835
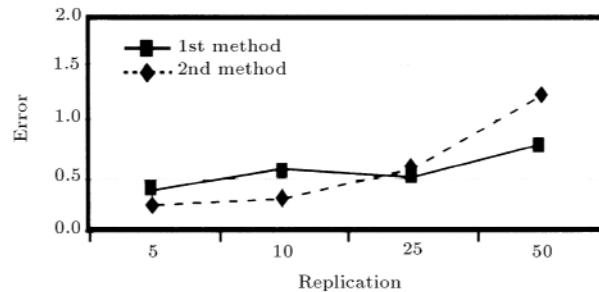
Figure 2. Graph of Class 1.

better than the 2nd method for a small number of replications and is worse for a large number of replications. Moreover, the figures show that the errors in both methods, when the number of replications is at its medium level, are relatively low. In this case, the second method performs better when both variability and order of example factors are set to their low or high levels. Besides, in cases when the variability factor is set to its low level, increasing the number of replications would result in increasing the variance of the error term and, hence, decreasing the power of the test statistics. Furthermore, with the variability factor set to its high level, increasing the number of replications would not much decrease the magnitude of the variance of the error term. It is recommended to do the experiments with an average number of replications.

In order to statistically compare the performances of the two methods, the methods are considered as the treatment levels. Since different random number
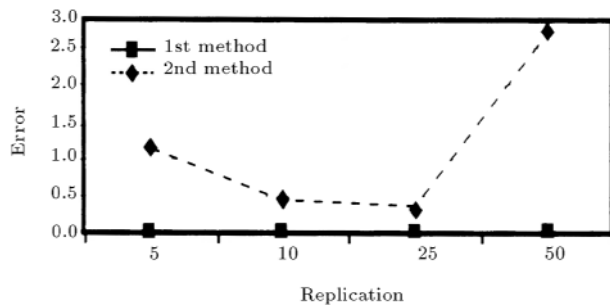
Table 6. Values of the F-test statistic.

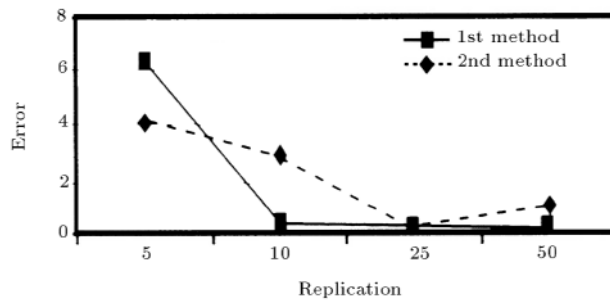| Type	Class 1	Class 2	Class 3	Class 4	Comprehensive Case
F_0	0.0902	0.43699	4.3966	0.089	4.099
p-value	≈ 0.428	≈ 0.495	≈ 0.192	≈ 0.527	≈ 0.291

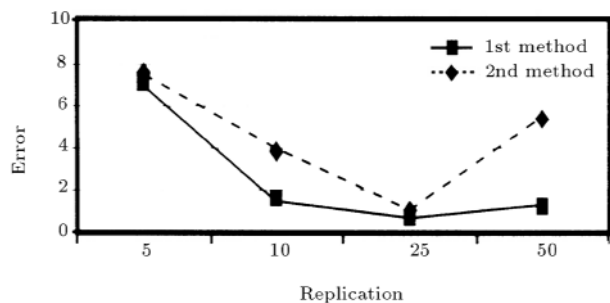$F_{0.05, 1.3} = 10.13$
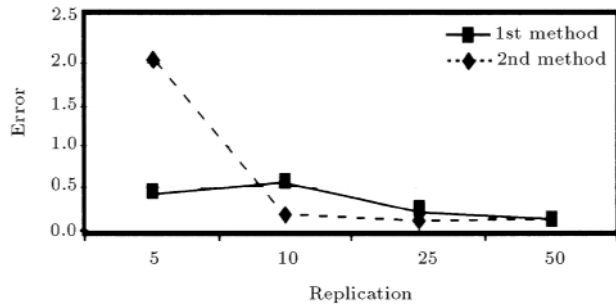
Figure 3. Graph of Class 2.

Figure 4. Graph of Class 3.

Figure 5. Graph of Class 4.

Figure 6. Graph of the comprehensive case.

seeds are used in different replications, the randomized complete block design [1] is used, the blocks being the replications. This design is applied to both of the classes and for the comprehensive case, and the performances of the methods are compared for each class. Thus, two treatments and four blocks are defined. The observations are the performances of the methods given in Tables 1 to 5. The linear statistical model of this experiment is:

$$Q_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij} \quad i = 1, 2, \; j = 1, 2, 3, 4, \quad (13)$$

where Q_{ij} is the performance or error value of the ith method in the jth replication, μ is an overall mean, τ_i is the effect of the ith method, β_j is the effect of the jth replication and ε_{ij} is the usual NID$(0, \sigma^2)$ random error term.

Table 6 summarizes the computational results of the experiment. From this table, one can see that the two GA methods are not statistically different in all of the cases.

Benchmark Case

For better comparison of the presented methods with similar works, in this section a benchmark example is used. This is a numerical example about a chemical process that was used by Cheng et al. [14] in their research. In this problem, there are three design variables and two responses. The design variables are reaction time (x_1), temperature (x_2) and percent catalyst (x_3) and the responses are percent conversion (y_1) and thermal activity (y_2). The objective is to maximize y_1 while keeping y_2 between 55 and 60, with a target value of 57.5.

Although the characteristics of the framework presented in Cheng et al.'s work is totally different from the ones in this research, in the sense that the data are the results of designed experiments (as opposed to simulated results) and the factor levels do not vary in a continuous manner (as opposed to real ones), for comparison purposes it was attempted to fit and use this example in our methodology.

Based on the experimental design data [14], two linear polynomial functions were estimated for each response. These estimates are:

$$y_1 = 78.3 + 1.0284x_1 + 4.0403x_2 + 6.2037x_3,$$

$$y_2 = 60.51 + 3.583x_1 + 0.2546x_2 + 2.2298x_3.$$

It is assumed that the error term has normal distribution with mean zero. Based on the experimental results, the variance of the error terms in each model were estimated and, hence, the distributions of the error terms were estimated as: $\varepsilon_1 \sim N[0, 33.2]$ and $\varepsilon_2 \sim N[0, 0.73066]$.

The one-sided desirability function was selected for the first response and its minimum and maximum values were set as 50 and 100, respectively. Also, a two-sided desirability function was selected for the second response and the values of 55, 60 and 57.5 were set as its minimum, maximum and target value, respectively. The possible ranges for x_1, x_2 and x_3 are set within the interval [2, 2]. The number of replications was set to be equal to 25. Table 7 summarizes the results obtained by the two proposed methods.

The results show that in terms of the first response value, the first method works better than the second method. However, in terms of the second response, the second method works better than the first one. Although the nature of this problem is not the same as the ones for the proposed method, the results are very close to the results of Cheng et al.'s research [14].

CONCLUSION AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper, a multi-response statistical optimization problem was modeled through the desirability function approach. Then, two GA methods were applied to solve this model by simulation. The performance of each method was studied through different simulation replications and statistically compared by a performance measure. The statistical design used for this comparison was the randomized complete block design, the blocks being four replications in each case. It was concluded that no statistical difference existed between the two methods in all situations. This was the case if the experiments were not performed in classes. However, the value of the test statistic becomes larger in the latter case. At the end, the performance of the proposed methods was compared with the one by Cheng et al.'s procedure through a designed experiment and it was shown that the proposed methods work well in experimental environments too.

For future research in this area, the following are recommended:

1. One needs to perform some experiments at the

Table 7. The results of the benchmark case.

Method	x_1	x_2	x_3	y_1	y_2
1st Method	-1.9408	1.9924	1.987	96.6793	58.4939
2nd Method	-1.9871	1.9973	1.9718	96.5572	58.2947

medium levels of the order and variability factors. This can be done either by a 2^3 design or the credibility of the statistical test can be enhanced by making some experiments at the center point of a 2^2 design;

2. In cases when the user has no idea how to set the parameters l and u in the initial step, their extreme values can be assigned. Then, the method can be applied sequentially, selecting the best solution at each step. Next, the solution is replaced with l in the maximization case and with u in a minimization environment. This process continues until no feasible solution is obtained;

3. In addition to Tuckey's test, other multiple comparison tests can be performed. The performance of other statistical tests can be compared in this regard;

4. Different crossover and mutation operators in the GA methods may lead to different conclusions;

5. One may think of different fitness functions in the algorithms;

6. Instead of selecting the chromosomes based on their fitness function values, they can be chosen randomly. This will lead to another type of research, when a comparison can be made between the two ways of chromosome selection;

7. Instead of preparing the data by simulation, the method can be applied to a real world situation, such as quality improvement in industry, where the best subset of the input variables are to be chosen for multi-response optimization purposes;

8. Rather than polynomials, the performance of the method can be evaluated by some other functions of the input variables;

9. A comprehensive set of benchmark cases is needed for comparison of the performance of different search-heuristic methodologies in this area.

REFERENCES

1. Montgomery, D.C., *Design and Analysis of Experiments*, Fourth Ed., John Wiley & Sons, New York, USA (1997).

2. Azadivar, F. "Simulation optimization methodologies", *Proceedings of the 1999 Winter Simulation Conference* (1999).

3. Coello Coello, C.A. "An updated survey of GA-based multi-objective optimization techniques", *ACM Computing Surveys*, **32**, pp 109-143 (2000).

4. Hartmann, N.E. and Beaumont, R.A. "Optimum compounding computer", *Journal of the Institute of the Rubber Industry*, **2**, pp 272-275 (1968).

5. Biles, W.E. and Swain, J.J. *Optimization and Industrial Experimentation*, Wiley Inter-Science, New York, USA (1980).

6. Box, G.E.P. and Liu, P.Y.T., *Empirical Model Building and Response Surface*, John Wiley & sons, New York, USA (1987).

7. Clayton, E.R., Weber, W.E. and Taylor, B.W. "A goal programming approach to the optimization of multi-response simulation models", *IIE Transactions*, **14**, pp 282-287 (1982).

8. Rees, L.P., Clayton, E.R. and Taylor, B.W. "Solving multiple response simulation optimization models using modified response surface methodology within a lexicographic goal-programming framework", *IIE Transactions*, **17**, pp 447-457 (1985).

9. Baseler, F.F. and Sepulveda, J.A. "Multi-response simulation optimization using stochastic genetic search within a goal-programming framework", *Proceedings of the 2000 Winter Simulation Conference* (2000).

10. Mollaghasemi, M., Evans, M.G. and Biles, W.E. "An approach for optimizing multiple response simulation models", In *Proceedings of the 13th Annual Conference on Computers in Industrial Engineering*, pp 201-203 (1991).

11. Mollaghasemi, M. and Evans, G.W. "Multi-criteria design of manufacturing systems through simulation optimization", *Transactions on Systems, Man, and Cybernetics*, **29**, pp 113-121 (1994).

12. Teleb, R. and Azadivar, F. "A methodology for solving multi-objective simulation optimization problems", *European Journal of Operational Research*, **72**, pp 135-145 (1994).

13. Boyle, C.R. "An interactive multiple response simulation optimization method", *IIE Transactions*, **14**, pp 453-463 (1996).

14. Cheng, B.C., Cheng, C.J. and Lee, E.S. "Neuro-fuzzy and genetic algorithm in multiple response optimization", *Computers and Mathematics with Applications*, **44**, pp 1503-1514 (2002).

15. Schaffer, J.D. "Multiple objective optimization with vector evaluated genetic algorithms", *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pp 93-100 (1985).

16. Allenson, R. "Genetic algorithms with gender for multi-function optimization", *EPCC-SS92-01*, University of Edinburgh, UK (1992).

17. Fourman, M.P. "Comparison of symbolic layout using genetic algorithms", *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pp 141-153 (1985).

18. Periaux, J., Sefrioui, M. and Mantel, B. "GA multiple objective optimization strategies for electromagnetic backscattering", *Engineering and Computer Science*, pp 225-243 (1996).

19. Coello Coello, C.A. "An empirical study of evolutionary techniques for multiobjective optimization in engineering design", Ph.D. Dissertation, Tulane University (1996).

20. Fonseca, C.M. and Fleming, P.J. "Genetic algorithm for multi-objective optimization: Formulation, discussion and generalization", *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp 416-423 (1993).

21. Kim, D. and Rhee, S. "Optimization of a gas metal arc welding process using the desirability function and the genetic algorithm", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **218**(1), pp 35-41 (2004).

22. Heredia-Langner, A., Montgomery, D.C., Carlyle, W.M. and Borror, C.M. "Model-robust optimal designs: A genetic algorithm approach", *Journal of Quality Technology*, **36**, pp 263-279 (2004).

23. Derringer, G. and Suich, R. "Simultaneous optimization of several response variables", *Journal of Quality Technology*, **12**, pp 214-219 (1980).

24. Del Castillo, E., Montgomery, D.C. and Mcarville, D.R. "Modified desirability functions for multiple response optimizations", *Journal of Quality Technology*, **28**, pp 337-345 (1996).

25. Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, USA (1989).

26. Gen, M., *Genetic Algorithm and Engineering Design*, John Wiley & Sons, New York, USA (1997).

27. Boesel, J., Nelson, B. and Ishii, N. "A framework for simulation optimization software", Technical Report, Dept. of Industrial Engineering and Management Sciences, Northwestern University (1999).