Using a Modified Genetic Algorithm to Find Feasible Regions of a Desirability Function

WEN WAN[1] and JEFFREY B. BIRCH[2]

[1] *Virginia Commonwealth University, Richmond, VA 23298-0032*
[2] *Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0439*

Abstract

The multi-response optimization (MRO) problem in response surface methodology is quite common in applications. Most of the MRO techniques such as the desirability function method by Derringer and Suich are utilized to find one or several optimal solutions. However, in fact, practitioners usually prefer to identify all of the near-optimal solutions, or all feasible regions, because some feasible regions may be more desirable than others based on practical considerations. In this paper, with benefits from the stochastic property of a genetic algorithm (GA), we present an innovative procedure using a modified GA (MGA), a computational efficient GA with a local directional search incorporated into the GA process, to approximately generate all feasible regions for the desirability function without the limitation of the number of factors in the design space. The procedure is illustrated through a case study. The MGA is also compared to other commonly used methods for determining the set of feasible regions. Using Monte Carlo simulations with two benchmark functions and a case study, it is shown that the MGA can more efficiently determine the set of feasible regions than the GA, grid methods, and the Nelder-Mead simplex algorithm.

KEY WORDS: Feasible Regions; Multi-response Optimization (MRO); Response Surface Methodology.

1 Introduction

In industry and many other areas of science, data collected often contain several responses (or dependent variables) of interest for a single set of explanatory variables (independent variables, or factors). It is relatively straightforward to find a setting of the explanatory variables that optimizes a single response. However, it is often difficult to find a setting that optimizes multiple responses simultaneously. Thus, a common objective is to find an optimal setting or several feasible settings of the explanatory variables that provide the best compromise when simultaneously considering the multiple responses. Such an objective is called the multi-response optimization (MRO) problem.

One of the most popular and formal approaches to MRO is to use some specific function (an objective function) to combine the responses so that the multiple dimensional problem can be transformed into a one-dimensional problem. Examples for such specific functions are the desirability function introduced by Derringer and Suich[1], the generalized distance measure function by Khuri and Conlon[2], and the weighted squared error loss function by Vining[3]. For each of these and other procedures, the objective functions could be highly nonlinear. Often, for each of these functions, there may exist several disjoint feasible operating regions for the MRO problem resulting in multiple local optima.

In fact, as stated in Myers et al[4], practitioners are usually interested in finding not only the optimal solution(s) but the near-optimal solutions as well. That is, the goal in MRO is to find all feasible regions defined as those locations of the factors that result in near-optimal responses. Identifying this set of feasible regions is often more useful to the practitioner than finding one or several optimal solutions, as certain feasible regions may be more desirable than others based on practical considerations. For example, some of the feasible regions may be larger than other feasible regions, and thus represent a broader range of operating conditions under which the process gives optimal or near-optimal results.

The approach of overlaying the response contour plots, as recommended in Myers et al[5], could be a choice to find the feasible operating regions by visual inspection. But this graphical approach is limited to two or three dimensional domains of explanatory variables or factors. Design-Expert (by Stat-Ease Inc.)[6] can find feasible regions by finding groupings of feasible points. Its underlying method is to use the Nelder-Mead simplex algorithm with many different starting points to collect feasible points. SAS also has a numerical optimizer which performs a grid search to find feasible regions. Both the simplex and grid methods can become computationally inefficient when applied to very complicated high dimensional objective functions because the simplex and grid approaches are local search algorithms.

We found that a genetic algorithm (GA), a global optimization tool, originally developed by Holland[7], could be a good choice for finding all feasible regions, even for very high dimensional, complicated problems. However, the GA approach is computationally intensive (Haupt and Haupt[8]). Typically a GA, in order to find feasible points, must evaluate an objective function a large number of times. For example, it may take many hours for the GA to perform a single evaluation of a complex objective function. In this case, the GA would become very time-consuming when determining the feasible regions.

In order to find as many feasible solutions as possible that consist of feasible regions and at the same time minimize the total number of function evaluations, we proposed a modified genetic algorithm (MGA) that incorporates the simplex algorithm, a local directional search method, into a traditional genetic algorithm (GA) process. By combining the GA with the simplex algorithm, the MGA utilizes the advantages of each. Specifically, the MGA combines the advantages of both the GA and simplex in that the GA can quickly find a near-optimal solution in a feasible region and the simplex algorithm can move around the near-optimal solution found by the GA procedure to collect more feasible points in that feasible region. In this study, we choose a desirability function as our objective function for the MRO problem. Our MGA approach can easily be extended to the other MRO objective functions previously mentioned.

We compare our MGA approach to three other methods for identifying feasible regions: the basic GA, the Nelder-Mead (NM) simplex algorithm, and the grid search method. Through a case study and our simulation studies, we show that the MGA is the most computational efficient method among the four methods in finding all the feasible regions for the desirability function approach.

The remainder of this paper is organized as follows. In section 2, we briefly introduce the desirability function method and the definition of its feasible regions. We give a brief introduction to the NM simplex, the grid search, the GA and MGA methods, respectively, in section 3. We then justify the MGA method as useful at approximating all feasible regions and present the procedure for using the MGA to approximate all feasible regions of the desirability function. In section 4, we illustrate the procedure through a case study showing that the procedure successfully identifies the disjoint feasible regions. Section 5 compares the four methods using the case study and two benchmark functions and concludes that the MGA is the most efficient method collecting as many feasible points as possible in a certain limited number of function evaluations.

2 Desirability Function and its Feasible Region

The desirability function method, proposed by Derringer and Suich[1], transforms each response into a dimensionless individual desirability value, scaled within $[0, 1]$, through a desirability function and then combines these individual desirabilities into one final desirability value using a geometric mean. The final desirability value (considered here as an objective function utilized for optimization) is given by

$$D = (d_1 \times d_2 \times \cdots \times d_m)^{1/m} \times 100\%, \tag{1}$$

where m is the number of responses and d_i is the i^{th} individual desirability values based on some appropriately chosen desirability function, a function of explanatory variables. The individual desirability functions are presented in the appendix. The goal in MRO is to find a setting of the explanatory variables that maximizes (an optimal solution) or nearly maximizes (a near-optimal solution) D. The range of D obviously is $[0, 1]$. Our goal is to identify all feasible regions which consists of all near-optimal solutions.

All feasible regions may be defined and constructed by all solutions which achieve $D \geq D_{cutoff}$, where D_{cutoff} is some appropriate value. The choice of D_{cutoff} may depend on the global maximum value of D, denoted by "D_{max}," and on the priority of the experiment. For example, suppose that the maximum value of D is $D_{max} = 0.85$. The feasible regions may be defined and constructed by all those feasible solutions which achieve $D \geq D_{cutoff} = 0.80$.

Four algorithms, presented in section 3, will be used to determine those sets of values of the explanatory variables satisfying the near-optimality condition $D \geq D_{cutoff}$. This set of points for each algorithm constitute the feasible regions for that algorithm. Each algorithm will require many iterations of its computational scheme to identify the feasible regions.

3 Brief Introduction to The Four Algorithms

3.1 Nelder-Mead Simplex Algorithm

The NM simplex method for finding a local maximum of a function of several variables has been devised by Nelder and Mead[9]. For two variables, a simplex is a triangle, and the method is a pattern search that compares function values at the three vertices of a triangle. The worst vertex, where $f(x, y)$ is smallest, is rejected and replaced with a new vertex. A new triangle is formed and the search is continued. The process generates a sequence of triangles (which might have different shapes), for which the function values at the vertices get smaller and smaller. The size of the triangles is iteratively reduced and the coordinates of the minimum point are found. The NM algorithm can easily be extended to higher dimensions. In many numerical tests, the NM algorithm succeeds in obtaining a good reduction in the function value using a relatively small number of function evaluations but it is easy for NM to converge to a local optimal and not suitable for a highly nonlinear objective function. The C++ code for NM is obtained from Numerical Recipes in C++ (William et al[10]).

3.2 Grid Search Algorithm

In the 1-dimensional parameter case, consider the optimization of a general function f. If we have n test points $x_1, ..., x_n$ such that $L \leq x_1 \leq ... \leq x_n \leq U$, then the set of values, $f(x_1), ..., f(x_n)$, gives some indication of the behavior of f. It is very straightforward to extend the idea to a higher-dimensional case. Usually the equal-distance grid on the region is considered. Also, usually the full grid search (when n is large) is impractical in higher dimensional problems. Even for a low dimensional problem, most of the function evaluations will be wasted.

3.3 Genetic Algorithm

In a GA, a search point, a setting in the search space, is coded into a string which is analogous to a *chromosome* in biological systems. The string (or chromosome) is composed of characters which are analogous to *genes*. In a response surface application, the chromosome corresponds to a particular setting of k explanatory variables in the design space, denoted by $\mathbf{x} = [x_1, x_2, ..., x_k]'$, where the i^{th} gene in the chromosome corresponds to a x_i, the value of the i^{th} regressor. In GA terminology, a set of multiple concurrent search points or a set of chromosomes (or individuals) is called a *population*. These chromosomes are evaluated for their performance with respect to an objective function. Each iterative step where a new population is obtained is called a *generation*.

The three main GA operations, utilized to generate a new population from the current population in each iterative step, consist of selection, crossover and mutation. Each GA operation involves randomness. The selection operation involves randomness in that two chromosomes are randomly selected from the current population and become a pair of parents to produce two new offspring. The crossover operation allows for the exchange of some information from a pair of parents and its transmission to next generation. The locations of the crossover points are randomly chosen. The mutation operation is used to alter a very small number of the "genetic material" in a random fashion, enhancing the diversity of the population. The locations where mutation occurs for each chromosome in the current population are also randomly chosen.

A GA process is as follows. The initial population, a collection of n chromosomes, or $\mathbf{x}_1$,...,$\mathbf{x}_n$, is randomly generated. Each chromosome in the initial population is evaluated in terms of an objective function. Then the initial population, which becomes the current parent population, generates an offspring population using the GA operations of selection, crossover, and mutation. The objective function is evaluated for each chromosome or each $\mathbf{x}$ in the offspring population. A decision is then made, based on the value of the objective function for each $\mathbf{x}$, as to which chromosomes from among the offspring population and the current parent population are retained into the next generation as a new population. This GA process will be iteratively continued until some convergence criterion of has been satisfied. More details on the GA can be seen, for example, in Holland[7] and Haupt and Haupt[8].

A basic GA procedure has the following steps.

1. Define an objective/fitness function, and its variables. Set GA operations (such as population size, parent/offspring ratio, selection method, number of crossovers and mutation rate).

2. Randomly generate the initial population. That is, obtain $\mathbf{x}_1$,...,$\mathbf{x}_n$.

3. Evaluate each chromosome in the initial population by the objective function. That is, compute $D(\mathbf{x}_1)$, ..., $D(\mathbf{x}_n)$, where $D(\mathbf{x}_i)$ is the desirability function evaluated at the i^{th} chromosome $\mathbf{x}_i$.

4. Generate an offspring population, by GA operations (such as selection/mating, crossover, and mutation). That is, generate a new set of n settings of the regressors, using the operations of selection, crossover, and mutation of $\mathbf{x}_1$,...,$\mathbf{x}_n$ to obtain $\mathbf{x}^*_1$,...,$\mathbf{x}^*_n$.

5. Evaluate each individual in the offspring population by the objective function.

6. Decide which individuals to include in the next population. This step is referred to as "replacement" in that individuals from the current parent population are "replaced" by a new population, whose individuals come from the offspring and/or parent population.

7. If a stopping criterion is satisfied, then the procedure is halted. Otherwise, go to Step 4.

3.4 A Modified Genetic Algorithm

The idea of the MGA is that during each GA iteration cycle, once the GA finds a feasible point whose D value is greater than D_{cutoff}, a NM simplex local search is implemented with that feasible point considered as a starting point. The local search is halted when the objective function fails to increase. During the MGA process, all the feasible points are recorded as long as their D values $> D_{cutoff}$. Our MGA combines the advantages of both the GA and NM methods. In the MGA process, the GA component moves throughout the regressor variable space to locate feasible points and the simplex component examines the local region around each feasible point to collect as many feasible points as possible in those feasible regions.

The MGA procedure is the same as that of GA, except that in the i^{th} generation we add step D between steps 5 and 6 in the original GA procedure as follows:

D. Is the best chromosome in the offspring population also the best over the current parent population and does the best offspring have a D value $> D_{cutoff}$? That is, is $\max(D(\mathbf{x}_i^*)) > D_{cutoff}$?

   D-1. If no, directly go to Step 6.

   D-2. If yes, then define and implement a NM local direction search with that chromosome as a starting point. The local search will be ended when the objective function fails to increase. Find the chromosome with the largest desirability value and replace the best chromosome with the largest desirability value in the offspring population with this point. Then go to Step 6.

3.5 Stopping Criterion

The grid method has no specific stopping criterion in that it will be halted when all the points on the uniform equal-distance grid are evaluated. The NM simplex method will be halted when either or both a certain number of iterations occurs or the fractional convergence tolerance, a very small value, is achieved. Due to the purpose of finding feasible regions instead of a single global optima, the GA method will be stopped when a certain number of iterations is achieved. The settings of the NM simplex procedure used in MGA including the stopping criterion are the same as the ones used in the simplex method alone, and the settings of a GA are the same as the ones of MGA.

3.6 A Simulation Study to Compare the Four Algorithms

Logic dictates that as the number of function evaluations increases so does the number of feasible points. It is conceivable, for example, that each procedure would be able to find all feasible points provided an unlimited number of function evaluations. A computationally efficient method will find many of the feasible points with less computational effort than another method that finds as many feasible points. Conversely, a method that finds more feasible points with the same number of function evaluations as another method is more computationally efficient and would be the preferred method. Due to the stopping criterion used for NM, as mentioned in section 3.5, we cannot control for equality of the number of function evaluations for the NM simplex method and therefore for the MGA method as well. Consequently, we consider the efficiency index, given by

$$\text{efficiency index} = \frac{\text{total number of feasible points collected}}{\text{total number of function evalautions}},$$

as a comparison measure of computational efficiency. The higher the efficiency index, the higher computational efficiency of a method for finding feasible regions.

To obtain a sufficient number of feasible points to establish dense feasible regions, each algorithm has to be repeated many times using different starting points. To compare the four algorithms via Monte Carlo simulations, we repeated the procedure for finding feasible regions 500 times for each algorithm. Then the mean of efficiency index (denoted by "EI") can be calculated by

$$\text{Mean(EI)} = \sum_{i=1}^{n} \text{EI}_i/500,$$

where $i = 1, ..., 500$, and its Monte Carlo error by

$$\text{MC error} = \sqrt{\frac{\sum_{i=1}^{n}(\text{EI}_i - \text{Mean(EI)})^2}{500-1}}/\sqrt{500}.$$

3.7 Procedure of using MGA to Find Feasible Regions

The NM, GA, and MGA procedures are designed to search out the locations of optimal points. However, since the desirability functions are often extremely nonlinear with possibly many local optima, it is necessary to repeat each procedure many times to adequately describe the feasible regions. Below we describe, using the MGA procedure, the steps required to obtain these feasible regions. These same steps are used for the NM and GA procedures as well.

The procedure of using MGA to determine the approximate feasible regions has four stages, listed as follows.

1. MGA is repeated several times (say, five times) with different random seeds and an optimum and its corresponding location are recorded for each repetition. The best optimum among all these recorded optima is considered as a global optimum.

2. Based on the results from Stage 1 and based on the priority from the experimenters, the feasible solutions of the desirability function D are determined. That is, an appropriate cutoff value, D_{cutoff}, is chosen so that all locations found by MGA which achieve corresponding D values greater than this cutoff are regarded as feasible points.

3. When the MGA is repeated many times with different random seeds and with different settings of the GA operations such as different crossover points and mutation rates, feasible points are collected. As the number of repetitions of MGA is increased, the approximate feasible regions approach the true feasible regions.

4. Plot these feasible points in pairwise 2-dimensional axes. Then, based on these plots, calculate the feasible regions for each factor.

4 Case Study: A Chemical Process

To illustrate finding the feasible regions for a specific problem, we consider the following example from Myers et al[5], where a central composite design was conducted for a chemical process. Two explanatory variables (or factors) are time (x_1) and temperature (x_2). Three responses of interest are yield (y_1), viscosity (y_2) and number-average molecule weight (y_3). The collected data are given in Myers et al[5]. As in Myers et al[5], we transform the natural independent variables into the coded variables within the range of [0,1].

In this case study, their MRO goal is to maximize y_1 (the minimum L = 70 and optimum T = 80), and achieve a target value for y_2 (the minimum L = 62, the target T = 65, and the maximum U = 68), and, at the same time, control y_3 within the acceptable range of [3200, 3400]. The desirability function method is utilized to find simultaneous optimum solutions of the responses y_1, y_2, and y_3. In addition, the solution vector, $\mathbf{x}_s$, should be within the experimental region R, which is defined as $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5^2$ in this case study.

4.1 Results by MGA Through the Procedure

In Stage 1, under the same conditions and fitted models given in Myers et al[5], the two solutions we found by MGA are listed as follows.

1)	$x_1 = 0.5767$	$x_2 = 0.1624$	$\hat{y}_1 = 78.6344$	$\hat{y}_2 = 65.0000$	$\hat{y}_3 = 3261.3111$	$D = 0.9292$
2)	$x_1 = 0.2676$	$x_2 = 0.7964$	$\hat{y}_1 = 78.2821$	$\hat{y}_2 = 65.0000$	$\hat{y}_3 = 3400.0000$	$D = 0.9101$

These two solutions are different from the two solutions obtained by Design-Expert as shown in Myers et al[5] (whose two values of D are 0.822 and 0.792) in terms of fitted optimal values for all of the three responses. The solutions obtained by MGA result in larger values of D, indicating that MGA performs better, in this example, at finding the optimal value of D than the Nelder-Mead simplex algorithm used by Design-Expert.

Figure 1 represents the surface (the left graph) of the desirability function D within the experimental region R and its corresponding contour plot (the right graph). The figure shows that there are two distinct surfaces which represent two disjoint operating regions. Obviously, the surface of D matches well to the contour plot. In addition, the two optimal solutions we found also match well to the figure.

[**Figure 1 here**]

Based on the results from Stage 1, the feasible solutions can be defined by choosing appropriate cutoff values in terms of the desirability function D. In this study, several cutoff values, 0.2, 0.5, 0.8 and 0.9, are used to check if MGA can determine the two feasible regions by collecting feasible points. That is, if the cutoff value of D (0.2, 0.5, 0.8 or 0.9) is achieved, then the corresponding location, which is regarded as a feasible point, is recorded during the MGA process. MGA with 100 iterations is repeated 20 times with 20 different starting random seeds and with 12 different settings of the GA operations to obtain a sufficiently large number of the feasible points. We note that during the MGA process, some of the same feasible points/locations may be found multiple times. The CPU time is only about 8 seconds using a moderately equipped PC.

Figure 2 represents the plots of the feasible points collected by MGA with different cutoff values (0.2, 0.5, 0.8, and 0.9) respectively. It shows that the observed feasible points define two disjoint regions, which correspond to the peaks of the two surfaces shown in Figure 1. With the cutoff values increasing from the left to the right in Figure 2, the regions become smaller and narrower. Compared to the contour plot of the desirability function in Figure 1, it is easy to see that the shapes and sizes of the two disjoint regions are very close to the ones of the contour plot at the four different levels of cutoff values of D. That is, the two disjoint regions are defined very well by the feasible points collected using MGA.

[**Figure 2 here**]

Based on the knowledge from the plots in Figure 2, we can calculate the feasible regions for each factor. For example, suppose that only values of D greater than or equal to 0.9 are acceptable. To calculate the approximate feasible regions, one feasible region would be x_1 in [0.247, 0.268] with x_2 in [0.795, 0.798] and the second feasible region would be x_1 in [0.460, 0.798] with x_2 in [0.153, 0.176]. Obviously, the second feasible region is larger than the first one in terms of the ranges of both factors x_1 and x_2. Therefore, the second feasible region would be considered to be more desirable than the first one, due to a broader set of the operating conditions. In addition, in the feasible region, the factor x_1 (time) is more robust than the factor x_2 (temperature), because x_1 has a wider range than x_2 to achieve the same feasible operating region. Note that the feasible regions calculated only give us the information on the upper and lower bounds for each factor. For more information about the feasible regions such as the shape, we have to rely on the plots of the feasible regions, approximately displayed by the collected feasible points.

In this case study, the surface of the desirability function D and its contour plot are utilized to check the performance of our method to identify all feasible regions. If the case study had more than two or three factors/dimensions, then it would be difficult to graphicly show the surface of D and its contour plot. Thus, in such situations, we could not tell where the optimal solution is and where the feasible regions are. However, we still could use MGA to find its optimal solution and all its feasible regions and plot them in pairwise 2-dimensional axes.

4.2 Comparison Results

Table 1 shows the results of comparisons of the grid, NM simplex, GA, and MGA methods via the Monte Carlo simulations, with the cutoff value 0.9, as mentioned in Section 3.6, in terms of mean total number of function evaluations, mean total number of feasible points collected, and mean efficiency index over 500 simulations. The Monte Carlo errors were provided in parentheses. We did not do simulations for the grid method because its function evaluations were performed on the uniform equal-distance grid. In Table 1, the averaged total number of function evaluations were different as was the mean number of feasible points found. The mean efficiency indexes were 0.02, 0.28, 0.40, and 0.42, for the grid, NM simplex, GA, and MGA procedures, respectively, and implies, for this example, that MGA is the most efficient method, followed by the GA and NM simplex methods. The grid procedure is extremely inefficient in terms of our EI criteria. In other words, our MGA method collected relatively more feasible points per number of function evaluations than the other methods.

[Table 1 here]

5 Two Benchmark Functions

In addition to the case study, the two benchmark functions, the Schwefel's and Rastrigin's functions, were used for the comparisons of the NM simplex, GA, and MGA methods. The grid method is not included due to its poor performance in the case study.

A generalized Schwefel's problem 2.26 from Schwefel[11] and Bartz-Beielstein[12], is given by

$$\sum_{i=1}^{k} -x_i \sin(\sqrt{|x_i|}), where - 500 \leq x_i \leq 500,$$

where k is the number of dimensions of the function. The minimum of the objective function is given by

$$\min(f(\mathbf{x})) = f(420.9687, ..., 420.9687).$$

The minimum is dependent on k, the number of dimensions. For example, if k =2, then the minimum value is -837.97. Figure 3 shows its 1- and 2-dimensional surfaces.

[Figure 3 here]

A generalized Rastrigin's function (Bartz-Beielstein[12]) is given by

$$f(\mathbf{x}) = \sum_{i=1}^{k} (x_i^2 - 10\cos(2\pi x_i) + 10), \text{ where} - 5.12 \leq x_i \leq 5.12, \tag{2}$$

where k is the number of dimensions of the function. Figure 4 shows its 1- and 2-dimensional surfaces. The surfaces are very bumpy in a narrow range [-5.12, 5.12]. The goal is to find a minimal value and its corresponding location. The minimum of this function is known as $\min(f(\mathbf{x})) = f(0, ..., 0) = 0$.

[Figure 4 here]

In this study, we consider the two-dimensional case (that is, $k = 2$) for the Schwefel's function and the three-dimensional case ($k = 3$) for the Rastrigin's function. Like the desirability function in the case study, each benchmark function has its feasible regions. Suppose a feasible point is defined and collected for feasible regions when the function satisfies ≤ -700 for the Schwefel's function or ≤ 3 for the Rastrigin's function. Figure 5 shows the feasible regions for the 2-dimensional Schwefel's function ≤ -700 in the left panel and for the 3-dimensional Rastrigin's function ≤ 3 in the x_1 vs x_2 design space in the right panel. The figures in the x_1 vs x_3 and x_2 vs x_3 design spaces for the Rastrigin's function are quite similar to the one in the right panel and not shown here because of the symmetric property of the Rastrigin's function for each dimension. We can see that there are three feasible regions in the Schwefel's function and nine in the Rastrigin's function.

[Figure 5 here]

Similar to the case study, we compare the NM simplex, GA, and MGA methods using the two functions. Table 2 shows the comparison results through the Monte Carlo simulations, as mentioned in Section 3.6, in terms of mean total number of function evaluations, mean total number of feasible points collected, and mean efficiency index over 500 simulations. The Monte Carlo errors were provided in parentheses. Obviously, MGA has the highest values of efficiency index in both benchmark functions, indicating that MGA is the most efficient method to collect feasible points.

[Table 2 here]

6 Conclusion

The multi-response optimization (MRO) problem in response surface methodology is quite common in applications. Most of the MRO techniques such as the desirability function method are utilized to find one or several optimal solutions. However, practitioners usually prefer to identify all of the near-optimal solutions, or all feasible regions, because some feasible regions may be more desirable than others based on practical considerations. The desirability function like other MRO methods is highly non-linear. In this study, based on a GA, we developed a modified GA, MGA, that combines the global optimization strategy of the GA with the local optimization characteristics of the Nelder-Mean simplex algorithm. Our MGA procedure quickly finds feasible regions, similar to the GA, and incorporates the Nelder Mead simplex algorithm to "walk" around in those feasible regions to collect as many feasible points as possible. We also present a procedure using the MGA to approximately generate all feasible regions for the desirability function without the limitation of the number of factors in the design space.

The MGA is also compared with the GA, Nelder-Mead simplex algorithm, and grid methods in terms of the efficiency index, using a case study and two benchmark functions through Monte Carlo simulations. A comparison of the results in Tables 1 and 2 show that the MGA procedure is the most computationally efficient among the four methods considered, in terms of the efficiency index, at identifying those feasible points belonging to each feasible region.

All the examples in the study are all low-dimensional problems. As the number of dimensional increase, the Nelder-Mead simplex and grid search methods both are less computationally efficient than the GA, the global optimization method, and much less than our MGA method. The procedure of finding feasible regions for a desirability function can easily be extended to finding feasible regions for other MRO techniques which have nonlinear objective functions such as the generalized distance measure function and weighted squared error loss function as mentioned in the introduction.

C++ code is available upon request from the authors.

References

[1] Derringer GC, Suich R. Simultaneous Optimization of Several Response Variables. *Journal of Quality Technology* 1980; 12: 214-219.

[2] Khuri AI, Conlon M. Simultaneous Optimization of Multiple Responses Represented by Polynomial Regression-Functions. *Technometrics* 1981; 23: 363-375.

[3] Vining GG. (1998). A Compromise Approach to Multi-response Optimization. *Journal of Quality Technology* 1998; 30: 309-313.

[4] Myers RH, Montgomery DC, Vining GG, Borror CM, Kowalski SM. Response Surface Methodology: A Retrospective and Literature Survey. *Journal of Quality Technology* 2004; 36(1): 53-77.

[5] Myers RH, Montgomery DC, Anderson-Cook C. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments.* Third Edition. John Wiley and Sons, Inc., New York, NY. 2009.

[6] Design-Expert. *Stat-Ease Inc.* Version 5. Minneapolis, MN. 1997.

[7] Holland JH. *Adaption in Natural and Artificial Systems: an Introduction Analysis with Applications to Biology, Control, and Artificial Intelligence* A Bradford Book: The MIT Press, Cambridge, Massachusetts, London, England. 1992.

[8] Haupt RL, Haupt SE. *Practical Genetic Algorithms.* John Wiley and Sons, Inc., New York, NY. 2004.

[9] Nelder JA, Mead R. A simplex method for function minimization. *Computer Journal* 1965; 7: 308313.

[10] William HP, Saul AT, William TV, Brian PF. *Numerical Recipes in C++: The Art of Scientific Computing.* Cambridge University Press. 2002.

[11] Schwefel HP. *Evolution and Optimum Seeking.* Sixth-Generation Computer Technology. New York NY: Wiley. 1995.

[12] Bartz-Beielstein T. *Experimental Research in Evolutionary Computation The New Experimentalism (Natural Computing Series).* Springer. 2006.

APPENDIX: Individual Desirability

The following details concern the individual desirability functions and their parameters.

As mentioned in section 2, $\hat{y}_i(\mathbf{x})$ is needed to transform into a dimensionless individual desirability, d_i, in the desirability function. There are two cases for transformation to consider: one-sided and two-sided transformations. One-sided transformations are used when the goal is to either maximize the response or minimize the response. Two-sided transformations are used when the goal is for the response to achieve some specified target value. When the goal is to maximize the i^{th} response, the individual desirability is given by the one-sided transformation

$$d_i = \begin{cases} 0 & \hat{y}_i(\mathbf{x}) < L \\ \left[\frac{\hat{y}_i(\mathbf{x})-L}{T-L}\right]^r & L \le \hat{y}_i(\mathbf{x}) \le T \\ 1 & \hat{y}_i(\mathbf{x}) > T \end{cases}, \tag{3}$$

where T represents an acceptable maximum value, L represents the acceptable minimum value and r is known as a “weight”, specified by the user. Similarly, when the goal is to minimize the i^{th} response, the corresponding individual desirability is written as the one-sided transformation

$$d_i = \begin{cases} 1 & \hat{y}_i(\mathbf{x}) < T \\ \left[\frac{U-\hat{y}_i(\mathbf{x})}{U-T}\right]^r & T \le \hat{y}_i(\mathbf{x}) \le U \\ 0 & \hat{y}_i(\mathbf{x}) > U \end{cases}, \tag{4}$$

where T is an acceptable minimum value and U is the acceptable maximum value.

When the goal is to obtain a target value, the individual desirability is given by the two-sided transformation

$$d_i = \begin{cases} 0 & \hat{y}_i(\mathbf{x}) < L \\ \left[\frac{\hat{y}_i(\mathbf{x})-L}{T-L}\right]^{r_1} & L \leq \hat{y}_i(\mathbf{x}) \leq T \\ \left[\frac{U-\hat{y}_i(\mathbf{x})}{U-T}\right]^{r_2} & T \leq \hat{y}_i(\mathbf{x}) \leq U \\ 0 & \hat{y}_i(\mathbf{x}) > U \end{cases} , \tag{5}$$

where T is the target value, and L and U are the acceptable minimum and maximum values respectively, and r_1 and r_2 are weights, specified by the users.

This desirability function D offers the user great flexibility in the setting of the desirabilities due to allowing users to chose appropriate values of L, U, and T, and of r, r_1, and r_2, for their different specific situations. For more details on the desirability function, see, for example, Derringer and Suich[1] and Myers et al[5].

Table 1: Comparisons of the grid, Nelder-Mead simplex, GA, and MGA methods through 500 Monte Carlo simulations, in terms of mean total number of function evaluations, mean total number of feasible points collected, and mean efficiency index. The MC errors are in parentheses.

	grid*	NM simplex	GA	MGA
Mean num of func eval	40000	37122.55 (31.53)	28800.00(0.00)	32627.36(26.20)
Mean num of feas points	956	10261.65 (10.88)	11371.17(32.15)	13652.26 (38.27)
Mean efficiency index	0.02	0.28 (2.68E-04)	0.40 (1.12E-03)	0.42 (1.12E-03)

*No simulations for the grid method

Table 2: Comparisons of the grid, Nelder-Mead simplex, GA, and MGA methods through 500 Monte Carlo simulations, in terms of averaged total number of function evaluations, averaged total number of feasible points collected, and averaged efficiency index. The MC errors are in parentheses.

	NM simplex	GA	MGA
	Schwefel function in 2-dimension		
Mean num of func eval	211661.02 (18.31)	163800.00 (0.00)	303700.87 (142.73)
Mean num of feas points	7104.95 (38.82)	35759.50 (16.18)	173358.91 (144.33)
Mean efficiency index	0.03 (1.8E-04)	0.22 (8.9E-05)	0.57 (2.2E-04)
	Rastrigin function in 3-dimension		
Mean num of func eval	26195.80 (93.24)	129893.58(27.84)	409505.54 (297.32)
Mean num of feas points	1134480.41 (25.15)	1082160.00(0.00)	1048832.70 (298.38)
Mean efficiency index	0.02 (8.9E-05)	0.12(4.5E-05)	0.39 (1.8E-04)

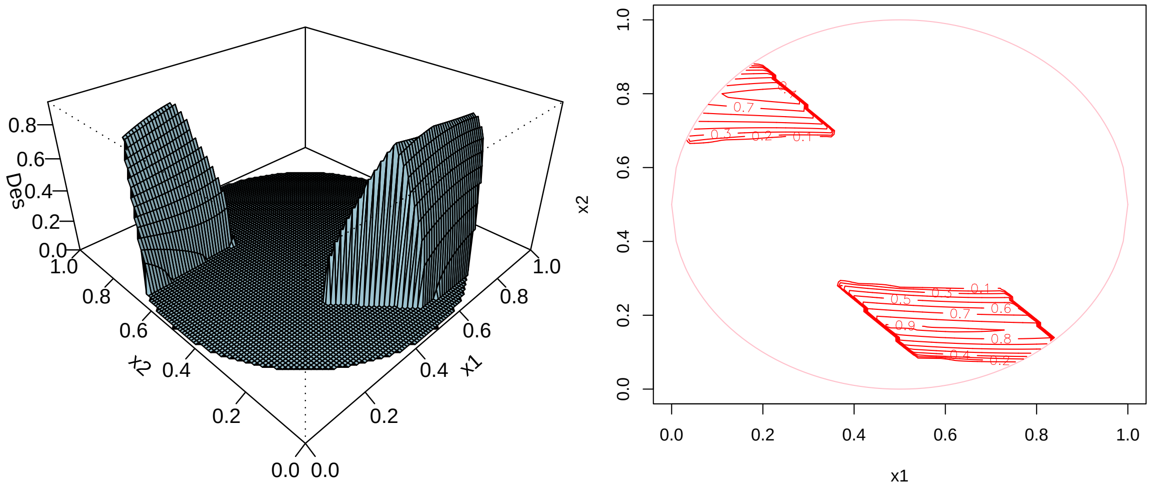

Figure 1: The 3-D surface and the contour plot of the desirability function (denoted by "Des") within the experimental region R in the case study of a chemical process: left: 3-D surface and right: contour

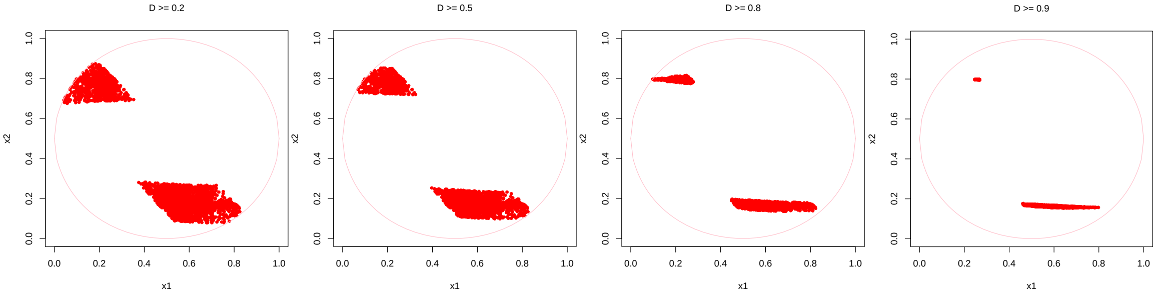

Figure 2: Plots of the feasible points collected by MGA with four different cutoff values in the case study of a chemical process: the first graph is by 0.2; the second is by 0.5; the third is by 0.8; and the last is by 0.9.

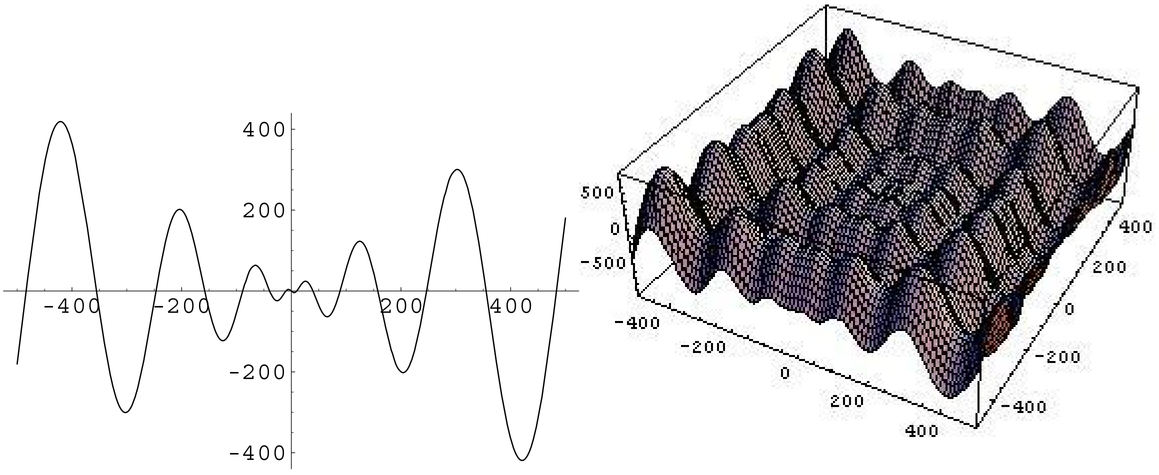

Figure 3: Surface of Schwefel's function. Left: 1-dimension; right: 2-dimension.

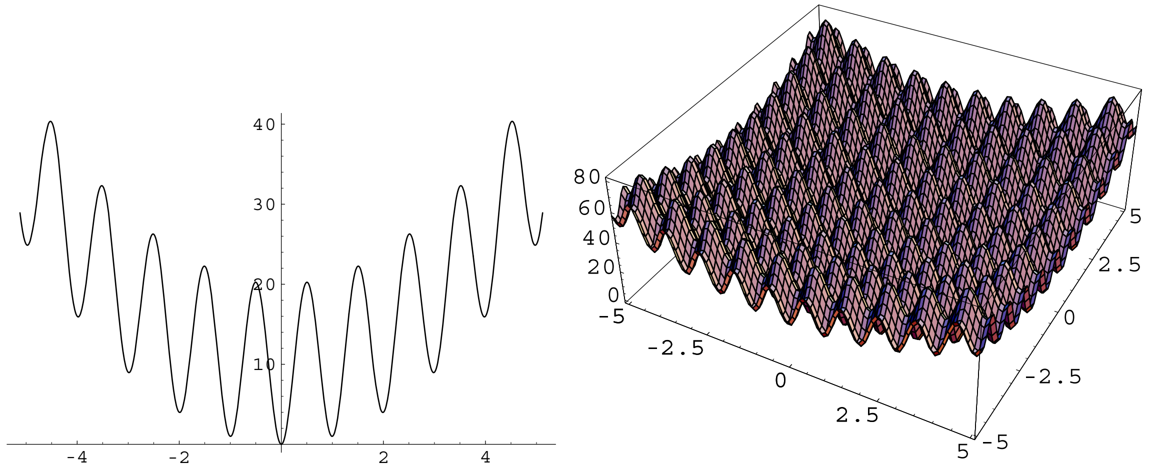

Figure 4: Surface of Rastrigin's function. Left: 1-dimension; right: 2-dimension.

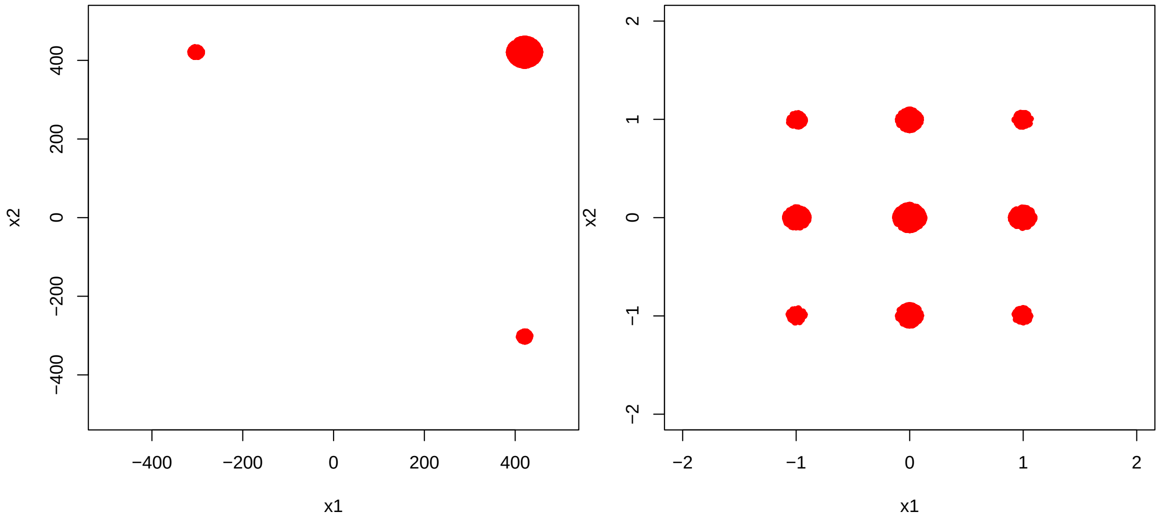

Figure 5: Plots of the feasible regions. Left: those satisfy ≤ -700 in the Schwefel's function; right: those satisfy ≤ 3 in the Rastrigin's function.

Authors' biographies

Wen Wan received her PhD in Statistics from the Virginia Polytechnic Institute and State University. She is an Assistant Professor in the department of Biostatistics at Virginia Commonwealth University.

Jeffrey B. Birch is Professor of Statistics and Director of Graduate Programs at Virginia Tech, where he has been a faculty member since 1977. Dr. Birch received his M.S. (1970) and Ph. D. (1977) degrees in biostatistics from the University of Washington. His area of primary research includes most aspects of regression analysis including robust regression, nonparametric and semiparametric regression, and profile monitoring. Dr. Birch is a former associate editor of Biometrics (1989-1993).