Thailand Statistician
January 2018; 16(1): 64-76
http://statassoc.or.th
Contributed paper

Using the GA Package in R Program and Desirability Function to Develop a Multiple Response Optimization Procedure in Case of Two Responses

Saranya Thongsook*
Department of Mathematics and Statistics, Faculty of Science and Technology, Pibulsongkram Rajabat University, Muang District, Phitsanuloke 65000, Thailand.
*Corresponding author; e-mail: s_thongsook@hotmail.com

Received: 18 November 2016
Accepted: 10 May 2017

Abstract

Multiple responses are one of the most frequently used methods in present experimental works. The desirability function is one popular approach for multiple response optimization (MRO). Many statistical software packages can also be used as optimization software for desirability function such as Design Expert, MINITAB, SAS, R and so forth. Most distributed software has more restrictive aspects governed by software licenses, while R is a free software that includes several optimization methods. The purpose of this research is to develop MRO procedure via the GA package in R and the desirability function to find a global optimization of multiple responses in case of two responses. The results from the GA package were compared with the results from the optim function, the rgenoud package, the DEoptim package and Design Expert (v.9 (Trial Version)) based on other algorithms to illustrate its performance. The results showed that the best D_Max values of three GA package methods were equal or superior to those values of other methods in R. Moreover, they were superior to those values of Design Expert in some cases. Therefore, it concluded that the GA package methods using built-in standard genetic operators in R and desirability function is a suitable method for finding a global optimization of multiple responses.

Keywords: Response surface methodology, statistical software packages, desirability function, R program.

1. Introduction

Response surface methodology (RSM) is a technique using not only to find functional relationship between several factors and a response for the purpose of predicting a future response, but also to decide whether values of factors are optimum in respect of the response. In present, RSM is applied for multiple responses (or dependent variables) of interest for a single set of explanatory variables (independent variables, or factors) in many fields, including engineering, agriculture and physical sciences.

RSM usually involves an experimental design, a response surface model and an optimization. For multiple response optimization (MRO), one of the approaches for MRO is to use some specific functions. Those functions combine the responses so that the multiple dimensional problem can be transformed into a one-dimensional problem. Examples of those functions are distance functions (Khuri and Conlon 1981), the square error loss functions (Pignatiello 1993, and Vining 1998), the desirability function (Del Castillo et al. 1996, Derringer and Suich 1980, and Harrington 1965). Furthermore, Ortiz et al. (2004) said that the desirability methods were easy to understand, implement, available in software, and provide flexibility in weighting individual responses.

Akteke-Ozturk (2010) mentioned that Design Expert, Excel Solver of Microsoft Office, SAS and MINITAB were optimization softwares for desirability function. However, those softwares were restricted as they were a licensed software. On the other hand, R was a free software that also included some built-in optimization algorithm. Scrucca (2013) summarized the functions and packages for an optimization in R. The optim, nlm, optimize function, the galts, mcga, rgenoud, genalg, DEoptim and GA package were examples of those functions and packages based on different algorithms. Moreover, his work showed that the GA package was more flexible than other functions and packages because users could defined their own objective function to be optimized. In 2015, Kuhn (2015) used the optim function that was not based on genetic algorithms and the desirability package in R to find a global optimization of multiple responses. However, other functions and packages in R have never been investigated in the global optimization of multiple responses.

Genetic algorithms have been successfully applied to solve optimization problems in many researches. The main advantage of genetic algorithms is an excellent method for solving a global optimization of problems. The GA package was one of the packages in R based on a genetic algorithm for solving optimization problems. Therefore, the purpose of this research was to develop MRO procedure via the GA package in R and the desirability function to find a global optimization of multiple responses in case of two responses. The MRO procedure will be reported. To illustrate the performances of the GA package based on genetic algorithms, their results are compared with the results of the optim function, the rgenoud package, the DEoptim package in R based on other algorithms. Moreover, the results of the GA package are also compared with Design Expert (v.9 (Trial Version)) which is a widely used program for an experimental design for more reliability.

2. Research Methodology

2.1. Desiability function approach

One of the most popular functions for multiple response optimization was introduced by Derringer and Suich (1980). They translated each response function into a desirability function and maximized the geometric mean of the desirability of each response by using single objective optimization technique. The desirability function form of each response for minimization and maximization cases shown in (1) and (2), respectively.

$$d_i = \begin{cases} 1 & \text{if } \hat{y}_i < T_i, \\ \left(\dfrac{U_i - \hat{y}_i}{U_i - T_i}\right)^{\varpi} & \text{if } T_i \le \hat{y}_i \le U_i, \\ 0 & \text{if } \hat{y}_i > U_i \end{cases} \tag{1}$$

$$d_i = \begin{cases} 0 & \text{if } \hat{y}_i < L_i, \\ \left(\dfrac{\hat{y}_i - L_i}{T_i - L_i}\right)^{\varpi} & \text{if } L_i \le \hat{y}_i \le T_i, \\ 1 & \text{if } \hat{y}_i > T_i \end{cases} \tag{2}$$

where L_i is the lower limit, U_i is the upper limit, T_i is the target value for response i and ω is the weight of the function. Then, the geometric mean of individual desirability is shown in (3).

$$D = \left(d_1^{r_1} \cdot d_2^{r_2} \cdot \ldots \cdot d_m^{r_m}\right)^{1/(r_1 + r_2 + \ldots + r_m)}, \tag{3}$$

where d_i is a desirability of each response and r_i is the weight or importance of each response.

2.2. The GA package in R (v. 3.2.3)

Genetic algorithms (GAs) were invented and developed by John Holland in the 1960s. Thereafter, GAs have been used to solve difficult optimization problems in many disciplines. For details of GAs and their applications, see Davis (1991), Michalewicz (1992), Haupt and Haupt (2004), and Sivanandam and Deepa (2008). Moreover, Mullen et al. (2011) mentioned that genetic algorithms had proven themselves to be useful heuristic methods for global optimization, in particular for combinatorial optimization problems. In 2013, Scrucca (2013), conceived of the GA package in R to solve optimal problems.

The GA package in R (v.3.2.3) implements a genetic algorithm for global optimization. The general steps of the genetic algorithm are shown in Figure 1.

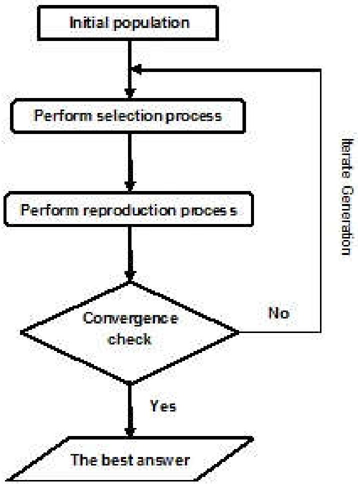


Figure 1 The general steps of the genetic algorithm

The GA package in R program was called “ga”, which had the following arguments:

```
ga(type = c("binary", "real-valued", "permutation"),
    fitness,min,max,nBits, population=gaControl(type)@population,
    selection = gaControl(type)@selection,
    crossover = gaControl(type)@crossover,
    mutation = gaControl(type)@mutation,
    popSize = 50, pcrossover = 0.8, pmutation = 0.1,
    elitism = max(1, round(popSize*0.05)), monitor = gaMonitor,
    maxiter = 100, run = maxiter, maxfitness = -Inf,
    names = NULL, suggestions, seed).
```

For using the GA package in this research, the variable type was set to be real-valued and the elitism was 2. The population sizes (N) were 10, 100 and 1000. The GA package was executed for 10000 generations for data. 36 different methods of the built-in standard genetic operators from the operator combinations of selection, crossover and mutation genetic operators (3 selections × 4 crossovers × 3 mutations) were applied (Table 1). The pcrossover and pmutation represent probability test values for the crossover and mutation operators, respectively. Each of these probability values were varied between 0-1.

Table 1 The built-in standard genetic operators in the GA package in R (v.3.2.3)

Selection genetic operator	Crossover genetic operator	Mutation genetic operator
linear-rank selection	single-point crossover	uniform random mutation
tournament selection	whole arithmetic crossover	nonuniform random mutation
truncation selection	local arithmetic crossover	random mutation around the solution
	blend crossover	

More detail on the underlying algorithms of the packages can be found in Scrucca or the GA package manuals in R (2008).

The real experiment for MRO was used in this paper. Studied of activated carbon on absorption of protein and sugar in protein solution extracted from defatted rice bran were used to verify our MRO procedure. These examples included two experimental designs, central compose designs (CCD) and near-optimal D small exact response surface designs (GA) proposed by Borkowski (2003). The sample sizes of CCD and GA were 30 (CCD), 28 (GA14) and 20 (GA10), respectively. Independent variables in this research were pH of the protein solution (x_1), incubation time (x_2) and activated carbon content (x_3). The amount of protein variable $\hat{y}_1$ (A280) and amount of sugar variable (A540) were determined. The goals of this study were maximization of the amount of protein and simultaneously minimized amount of sugar. The second-order model was applied as the predicted model for three trials as shown in (4).

$$\hat{y}_k(x) = \hat{\beta}_0 + \sum_{i=1}^{3} \hat{\beta}_i x_i + \sum_{i<j}^{3}\sum \hat{\beta}_{ij} x_i x_j + \sum_{i=1}^{3} \hat{\beta}_{ii} x_i^2, \tag{4}$$

where $\hat{y}_1(x)$ is the predicted amount of protein, $\hat{y}_2(x)$ is the predicted amount of sugar, $\hat{\beta}_0$ is a constant, $\hat{\beta}_i$ are the estimate of coefficient of pH of protein solution, incubation time and activated carbon content, respectively, $\hat{\beta}_{ij}$ are the estimate of coefficient for the interaction terms, and $\hat{\beta}_{ii}$ are the estimate of coefficient for the second-order terms.

The results of three trials were analyzed by R to find the predicted models for CCD, GA14 and GA10 shown as Models A, B and C, respectively. The three models obtained from three differently experimental designs so that the trial points and predicted models were different. Consequently, those models were applied to investigate the efficiency of the GA package for finding a global optimization of two responses.

$$\left.\begin{array}{l}\hat{y}_{1A}=80.299+2.14x_1-3.895x_2-5.975x_3-0.931x_1x_2+2.731x_1x_3-7.893x_2x_3-8.161x_1^2+10.164x_2^2+1.564x_3^2\\ \hat{y}_{2A}=64.924-4.875x_1-7.205x_2-17.29x_3-0.588x_1x_2-1.738x_1x_3-2.638x_2x_3-6.881x_1^2+6.619x_2^2+9.094x_3^2\end{array}\right\}(A)$$

$$\left.\begin{array}{l}\hat{y}_{2B}=90.343+4.942x_1-5.642x_2-11.593x_3+0.176x_1x_2+0.049x_1x_3-4.08x_2x_3-2.635x_1^2+3.673x_2^2+1.276x_3^2\\ \hat{y}_{2B}=69.889-0.87x_1-5.627x_2-18.979x_3+0.357x_1x_2-0.428x_1x_3-3.584x_2x_3-3.372x_1^2+2.728x_2^2+5.956x_3^2\end{array}\right\}(B)$$

$$\left.\begin{array}{l}\hat{y}_{1C}=90.475+7.928x_1+0.741x_2-7.424x_3+6.042x_1x_2+7.67x_1x_3+0.119x_2x_3-6.035x_1^2+2.94x_2^2-0.097x_3^2\\ \hat{y}_{2C}=74.302-0.038x_1+1.18x_2-11.744x_3+12.966x_1x_2+11.492x_1x_3+2.199x_2x_3-5.017x_1^2+6.42x_2^2+4.29x_3^2\end{array}\right\}(C)$$

Thereafter, the GA package and desirability function were used to find the global optimization for two responses. The GA package process was illustrated in Figure 2.

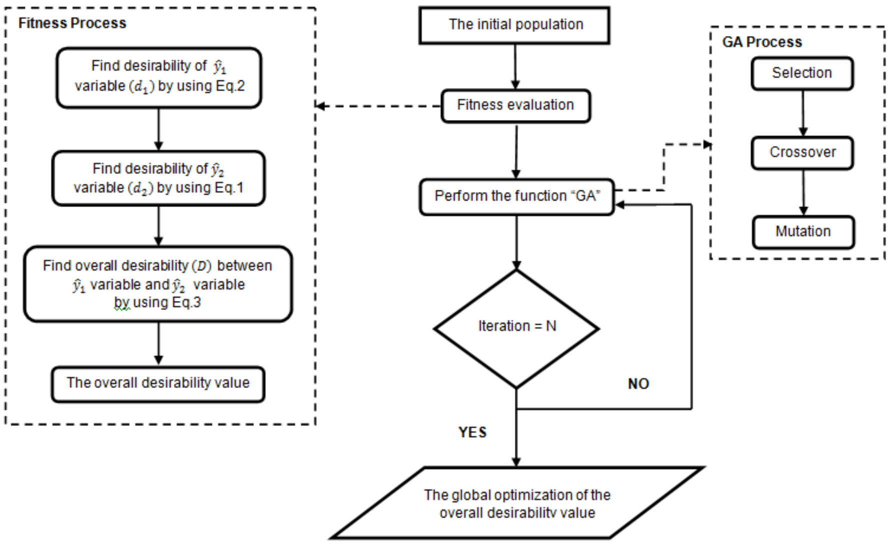

Fitness Process
Find desirability of $\hat{y}_1$ variable (d_1) by using Eq.2
Find desirability of $\hat{y}_2$ variable (d_2) by using Eq.1
Find overall desirability (D) between $\hat{y}_1$ variable and $\hat{y}_2$ variable by using Eq.3
The overall desirability value
The initial population
Fitness evaluation
Perform the function "GA"
Iteration = N
NO
YES
The global optimization of the overall desirability value
GA Process
Selection
Crossover
Mutation

Figure 2 The process of GA package in R

3. Results

The overall desirability value was maximized to obtain a global optimization for two responses. The maximization equation was exhibited below:

$$D\_Max = max(d_1^{r_1}\cdot d_2^{r_2})^{1/(r_1+r_2)}, \tag{5}$$

Results of three models (Models A, B and C) will be presented. Moreover, the results obtained from the GA package methods, the optim function, the rgenoud package, the DEoptim package and Design Expert (v.9 (Trial Version)) will be compared in terms of the best D_Max

values and the operating time. Specifying the value of the best D_Max was iteratively searched for the highest value until such searches D_Max value had no further improvement.

From 36 GA package methods, there were only three GA package methods which had the best D_Max values of all models. A selection, crossover and mutation operators for the first method, called GA_Method1, were linear-rank selection, blend crossover and nonuniform random mutation, respectively. Tournament selection, blend crossover and nonuniform random mutation operators were respectively used for the second method called GA_Method2. For the third method called GA_Method3, the operators were respectively included truncation selection, blend crossover and nonuniform random mutation.

For three GA package methods, the best D_Max values of Models A, B and C were 0.7468608, 0.7471575 and 0.7626133, respectively. However, the probability test values of the pcrossover and pmutation operators for each method were different. The probability values of both operators in GA_Method1, GA_Method2 and GA_Method3 were varied between 0.4000-0.9999, 0.7001-0.9900 and 0.1999-0.9091 across generations, respectively. Each GA package method was respectively executed for 10000 generations for Models A, B and C before changing the pcrossover and pmutation values. Table 2 respectively showed the best pcrossover and pmutation values for the best D_Max values in each population size and iteration.

Table 2 Summarize of the setting values and best values for all methods

Method	Model	Iteration	Population size	The best D_Max values	The probability of	
				D	crossover	mutation
GA_Method1	A	10,000	10,100,1000	0.7469	0.9001	0.9001
	B	10,000	10,100,1000	0.7472	0.7001	0.7001
	C	10,000	10,100,1000	0.7626	0.7001	0.7001
GA_Method2	A	10,000	10	0.7469	0.7500	0.7500
			100,1000	0.7469	0.7500	0.7500
	B	10,000	10	0.7125	0.9500	0.9500
			100,1000	0.7472	0.9500	0.9500
	C	10,000	10,100,1000	0.7626	0.8900	0.8900
GA_Method3	A	10,000	10,100,1000	0.7469	0.7999	0.7999
	B	10,000	10,100	0.7472	0.7901	0.7901
			1000	0.7472	0.7901	0.7901
	C	10,000	10,100	0.7626	0.5001	0.5001
			1000	0.7626	0.5001	0.5001

Figures 1, 2 and 3 showed the operating time for Models A, B and C in each method. The GA_Method2 took the longest operation time to achieve the best D_Max values. Moreover, the program execution time of the GA_Method1 was slightly different from those of the GA_Method3.

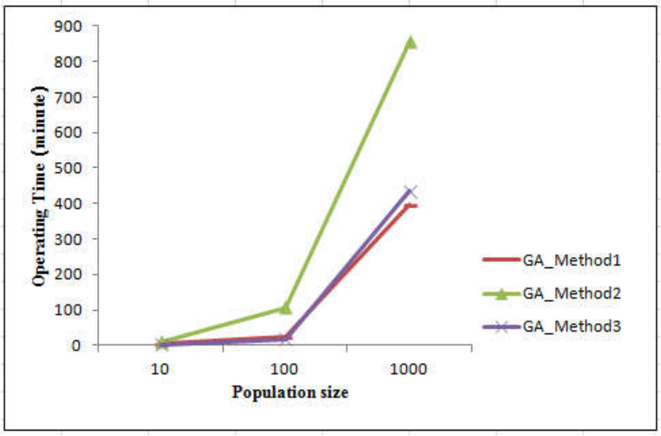


Figure 1 The operating times for Model A

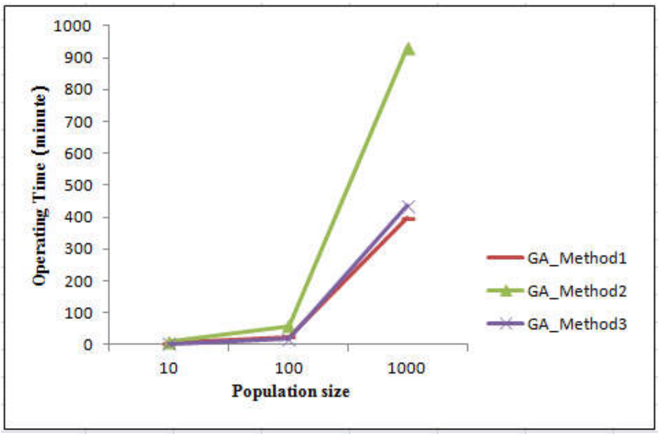


Figure 2 The operating times for Model B

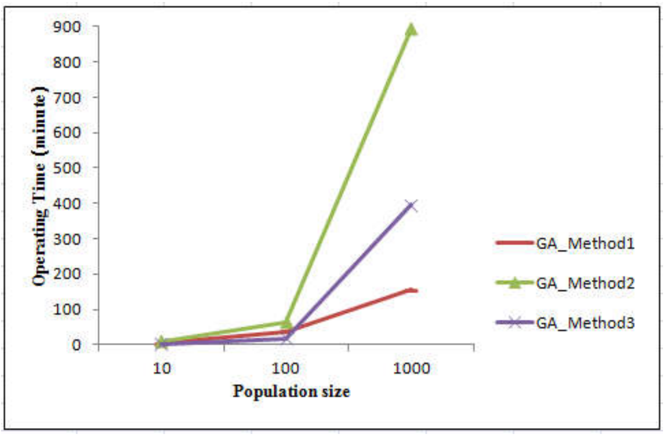
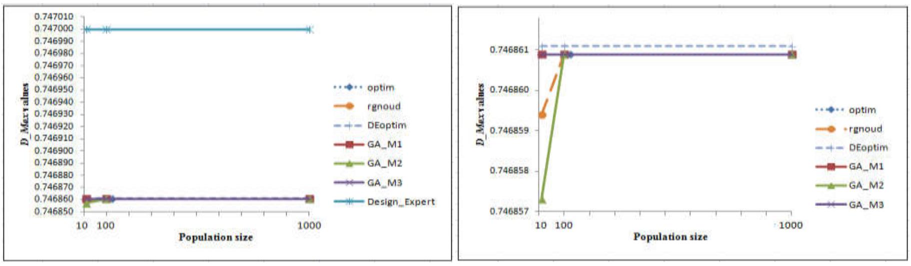
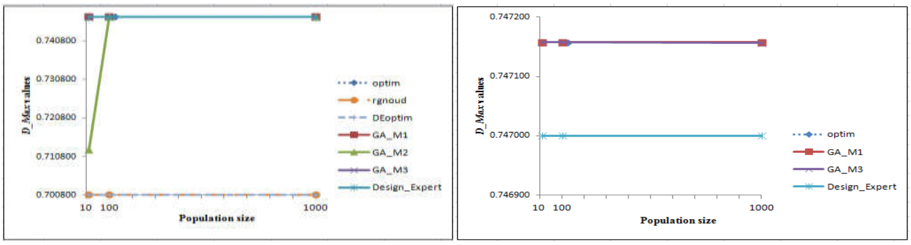


Figure 3 The operating times for Model C

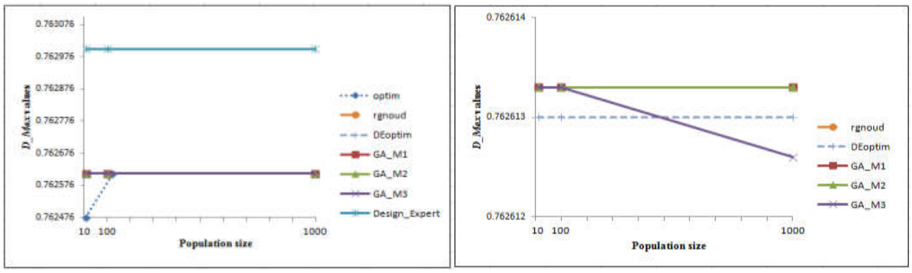
The best D_Max values of GA package methods, the optim function, the rgenoud, the DEoptim package in R and Design Expert v.9 (Trial Version) of Models A, B and C were compared as shown in Figure 4. For Model A, the results indicated that the best D_Max values of the GA_Method1 and GA_Method3 were slightly less than those values of the DEoptim package and Design Expert. However, they were equal to those values of the optim function. Moreover, the best D_Max values of the GA_Method2 and the rgenoud package were equal to those values of the GA_Method1 and GA_Method3 when the population sizes were 100 and 1000. The best D_Max values of the GA_Method1 and GA_Method3 were superior to those values of the rgenoud, the DEoptim package, Design Expert and equal to those values of the optim function for Model B. Moreover, the best D_Max values of the GA_Method2 were equal to those values of the GA_Method1 and GA_Method3 when the population sizes were 100 and 1,000. In case of Model C, the best D_Max values of the GA_method1 and GA_Method2 were equal to those values of the rgenoud package, slightly less than those values of Design Expert, but slightly greater than those values of the DEoptim package. Moreover, the best D_Max values of the GA_Method1 and GA_Method2 were also equal to those values of the optim function when the population sizes were 100 and 1000. However, the best D_Max values of the GA_Method3 were equal to those values of the GA_Method1 and GA_Method2 when the population sizes were 10 and 100.



(a) The best D_Max values of Model A



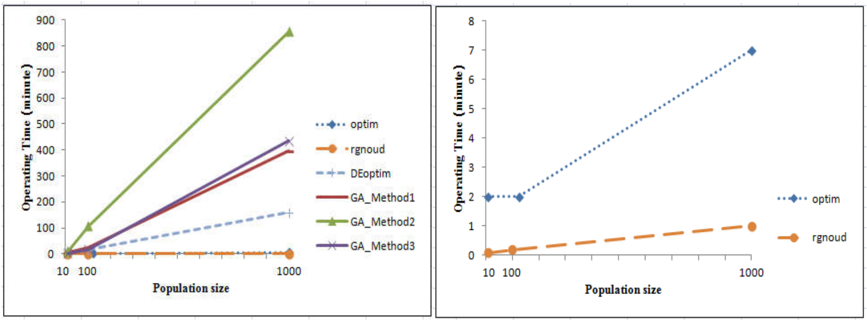
(b) The best D_Max values of Model B



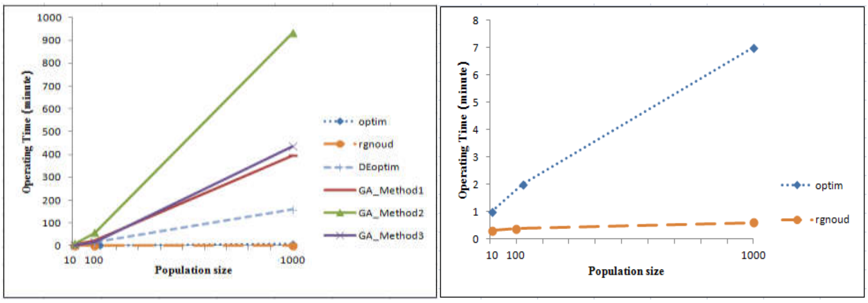
(c) The best D_Max values of Model C

Figure 4 Comparing the best values of all methods

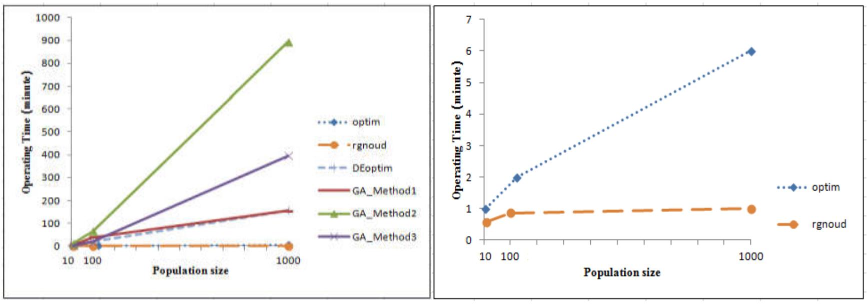
Considering the comparison of operating times in all methods, the results indicated that the GA_Method2 took the longest time to achieve the best D_Max values for all models. Moreover, the operating time of three GA package methods spent more time than other methods (Figure 5).



(a) The operating time of the Model A



(b) The operating time of the Model B



(c) The operating time of the Model C

Figure 5 Comparing the operating times of all methods

4. Conclusions

The primary goal of this research is to develop MRO procedure via the GA package by using built-in standard genetic operators in R and desirability function to find a global optimization of multiple responses in case of two response. The results showed that the best D_Max values of the GA package methods were equal or superior to those values of two packages and the function in R. Moreover, they were superior to those values of Design Expert in some cases. Although the GA package methods spent the operating time more than other methods, their processes worked very effectively on the best D_Max values. Therefore, it concluded that the GA package methods by using built-in standard genetic operators in R and desirability function is a suitable method for finding a global optimization of multiple responses.

Acknowledgements

The research was supported by Research and Development Institution, Pibulsongkram Rajabhat University.

References

Akteke-Ozturk B. New Approaches to desirability function by nonsmooth and nonlinear optimization. Ph.D [dissertation]. Turkish: Middle East Technical University; 2010.

Borkowski JJ. Using a genetic algorithm to generate small exact response surface designs. JPSS. 2003; 1: 65-88.

Davis LD. Handbook of genetic algorithms. New York: Van Nostrand Reinhold; 1991.

Castillo ED, Montgomery DC, McCarville DR. Modified desirability functions for multiple response optimizations. J. Qual. Technol. 1996; 28(3): 337-345.

Derringer G, Suich R. Simultaneous optimization of several response variables. J. Qual. Technol. 1980; 12(4): 214-219.

Harrington EC. The desirability function. Ind. Qual. Control. 1965; 21(10): 494-498.

Haupt RL, Haupt SE. Practical genetic algorithms. New York: John Wiley & Sons; 2004.

Khuri, AI, Conlon M. Simultaneous optimization of multiple responses represented by polynomial regression functions. Technometrics. 1981; 23(4): 363-375.

Kuhn M. The desirability package. 2015 [cited 2016 Jan 9]; 1-17. Available from URL: https://cran.r-project.org/web/packages/desirability/vignettes/desirability.pdf.

Michalewicz Z. Genetic algorithms + Data structures = Evolution programs. New York: Springer-Verlag; 1992.

Mullen K, David A, Ardia D, Gil D, Windover D, Cline J. DEoptim: an R package for global optimization by differential evaluation. J. Stat. Softw. 2011; 40(6): 1-26.

Ortiz F, Simpson JR, Pignatiello JJ, Heredia LA. A genetic algorithm approach to multiple-response optimization. J. Qual. Technol. 2004; 36(4): 432-450.

Pignatiello JJ. Strategies for robust multiresponse quality engineering. IIE trans. 1993; 25(3): 5-15.

Scrucca L. GA: a package for genetic algorithms in R. J. Stat. Softw. 2013; 53(4): 1-37.

Sivanandam SN, Deepa SN. Introduction to genetic algorithms. New York: Springer Press; 2008.

The R Development Core Team. R: A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing; 2008.

Vining GG. A compromise approach to multi-response optimization. J. Qual. Technol. 1998; 30(3): 309-313.

Appendix

For Model 1

```
>gaControl("real-valued")
>gaControl("real-valued"=list(selection = "gareal_sigmaSelection"))
>gaControl("real-valued"=list(crossover = "gareal_blxCrossover"))   % Choose selection, crossover, mutation
>gaControl("real-valued"=list(mutation = "gareal_rsMutation"))        genetic operators
> rsmOpt <- function(x, CCDd1, CCDd2) {
 + CCDd1 <- dMax(59.5381,100)
 + CCDd2 <- dMin(36.7511,100)
 + A280CCDPred <- 80.299+2.14*x[1]-3.895*x[2]-5.975*x[3]-0.931*x[1]*x[2]
                    +2.731*x[1]*x[3]-7.893*x[2]*x[3]-8.161*x[1]^2
                         +10.164*x[2]^2+1.564*x[3]^2
 + A540CCDPred <- 64.924-4.875*x[1]-7.205*x[2]-17.29*x[3]-0.588*x[1]*x[2]
                         -1.738*x[1]*x[3]-2.638*x[2]*x[3]-6.881*x[1]^2+6.619*x[2]^2
                         +9.094*x[3]^2                                          % The desirability
 + outCCDd1 <- predict(CCDd1, data.frame(A280CCDPred=A280CCDPred))                function to find
 + outCCDd2 <- predict(CCDd2, data.frame(A540CCDPred=A540CCDPred))                the best D_Max values
 + outCCD.D <- (outCCDd1*outCCDd2)^(1/2)
 + if(any(abs(x) > 1)) {
 +    outCCD.D <- 0
 + }
 + return(outCCD.D)
 + }
> format(Sys.time(), "%H:%M:%OS3")
> CCDGA <- ga(type = c("real-valued"), fitness = rsmOpt, min = c(-1, -1, -1), max = c(1, 1, 1),     %GA package
         popSize = 100, pcrossover = 0.6999, pmutation = 0.6999, elitism = 2, maxiter = 10000)
> format(Sys.time(), "%H:%M:%OS3")
>summary(CCDGA)
```

For Model 2

```
>gaControl("real-valued")
>gaControl("real-valued"=list(selection = "gareal_sigmaSelection"))
>gaControl("real-valued"=list(crossover = "gareal_blxCrossover"))   % Choose selection, crossover, mutation
>gaControl("real-valued"=list(mutation = "gareal_rsMutation"))        genetic operators
> rsmOpt <- function(x, GA14d1, GA14d2) {
+  GA14d1 <- dMax(63,100)
+  GA14d2 <- dMin(46.037,98)
+  A280GA14Pred <- 90.343+4.942*x[1]-5.642*x[2]-11.593*x[3]+0.176*x[1]*x[2]
                         +0.049*x[1]*x[3]-4.08*x[2]*x[3]-2.635*x[1]^2+3.673*x[2]^2
                         -1.276*x[3]^2
+  A540GA14Pred <- 69.889-0.87*x[1]-5.627*x[2]-18.979*x[3]+0.357*x[1]*x[2]
                         -0.428*x[1]*x[3]-3.584*x[2]*x[3]-3.372*x[1]^2+2.728*x[2]^2
                         +5.956*x[3]^2                                          % The desirability
+  outGA14d1 <- predict(GA14d1, data.frame(A280GA14Pred=A280GA14Pred))            function to find
+  outGA14d2 <- predict(GA14d2, data.frame(A540GA14Pred=A540GA14Pred))            the best D_Max values
+  outGA14.D <- (outGA14d1*outGA14d2)^(1/2)
+  if(any(abs(x) > 1)) {
+    outGA14.D <- 0
+  }
+  return(outGA14.D)
+ }
> format(Sys.time(), "%H:%M:%OS3")
> GA14GA <- ga(type = c("real-valued"), fitness = rsmOpt, min = c(-1, -1, -1), max = c(1, 1, 1),    %GA package
          popSize = 100, pcrossover = 0.6901, pmutation = 0.6901, elitism = 2, maxiter = 10000)
> format(Sys.time(), "%H:%M:%OS3")
>summary(GA14GA)
```

For Model 3

```
>gaControl("real-valued")
>gaControl("real-valued"=list(selection = "gareal_sigmaSelection"))
>gaControl("real-valued"=list(crossover = "gareal_blxCrossover"))    % Choose selection, crossover, mutation
>gaControl("real-valued"=list(mutation = "gareal_rsMutation"))         genetic operators
> rsmOpt <- function(x, GA10d1, GA10d2) {
+   GA10d1 <- dMax(63.2961,100)
+   GA10d2 <- dMin(46.037,97.4055)
+   A280GA10Pred <- 95.475+7.928*x[1]+0.741*x[2]-7.424*x[3]+6.042*x[1]*x[2]
                                        +7.67*x[1]*x[3]+0.119*x[2]*x[3]-6.035*x[1]^2+2.94*x[2]^2
                                            -0.097*x[3]^2
+   A540GA10Pred <- 74.302-0.038*x[1]+1.18*x[2]-11.744*x[3]+12.966*x[1]*x[2]
                           +11.492*x[1]*x[3]+2.199*x[2]*x[3]-5.017*x[1]^2+6.415*x[2]^2
                                            +4.29*x[3]^2                              % The desirability
+   outGA10d1 <- predict(GA10d1, data.frame(A280GA10Pred=A280GA10Pred))                 function to find
+   outGA10d2 <- predict(GA10d2, data.frame(A540GA10Pred=A540GA10Pred))                 the best D_Max values
+   outGA10.D  <- (outGA10d1*outGA10d2)^(1/2)
+   if(any(abs(x) > 1)) {
+     outGA10.D <- 0
+   }
+   return(outGA10.D)
+ }
> format(Sys.time(), "%H:%M:%OS3")
> GA10GA <- ga(type = c("real-valued"), fitness = rsmOpt, min = c(-1, -1, -1), max = c(1, 1, 1),     %GA package
           popSize = 100, pcrossover = 0.6001, pmutation = 0.6001, elitism = 2, maxiter = 10000)
> format(Sys.time(), "%H:%M:%OS3")
>summary(GA10GA)
```