

A Hybrid Global Optimization Method Based on Genetic Algorithm and Shrinking Box

Saleem Z. Ramadan¹

¹ Department of Mechanical and Industrial Engineering, Applied Science Private University, Shafa Badran, Amman, Jordan

Correspondence: Saleem Z. Ramadan, Department of Mechanical and Industrial Engineering, Applied Science Private University, Shafa Badran, 11931, Amman, Jordan. Tel: 962-799-110-835. E-mail: s_ramadan@asu.edu.jo

Received: September 14, 2015

Accepted: December 8, 2015

Online Published: January 7, 2016

doi:10.5539/mas.v10n2p67

URL: <http://dx.doi.org/10.5539/mas.v10n2p67>

Abstract

This paper proposes a hybrid genetic algorithm method for optimizing constrained black box functions utilizing shrinking box and exterior penalty function methods (SBPGA). The constraints of the problem were incorporated in the fitness function of the genetic algorithm through the penalty function. The hybrid method used the proposed Variance-based crossover (VBC) and Arithmetic-based mutation (ABM) operators; moreover, immigration operator was also used. The box constraints constituted a hyperrectangle that kept shrinking adaptively in the light of the revealed information from the genetic algorithm about the optimal solution. The performance of the proposed algorithm was assessed using 11 problems which are used as benchmark problems in constrained optimization literatures. ANOVA along with a success rate performance index were used to analyze the model.

Based on the results, we believe that the proposed method is fairly robust and efficient global optimization method for Constrained Optimization Problems whether they are continuous or discrete.

Keywords: optimization, genetic algorithm, shrinking search space, penalty function, black-box functions, constrained optimization problem

1. Introduction

Global optimization for Black-box functions, BBF, is still demanding even with the great advances in computational power of the modern computers as it is used in many applications including finite element analysis and computational fluid dynamics. Metamodels-based design optimization MBDO has been used for over two decades for optimizing such functions as they are using approximate models to optimize the BBF (Abu Bakar, Ramlan, Muhammad (2007), Bazaraa, Sherali, Shetty (2006), Belegundu, Arora (1985), Mahdavi, Fesanghary, Damangir (2007), Mallipeddi, Suganthan (2010)). Fitting response surfaces has been also used extensively in the literature to carry out MBDO (Camp (1955), Carroll & Fiacco (1961), Picheny, Wagner, Ginsbourger (2012), Pierskalla (1968), Ramadan, S. & Ramadan, K. (2012), Regis & Shoemaker (2005)) such that the response surfaces were used to visualize the relationship between the values of the variables and the objective function values. This information was then used to estimate the location of the optimal value. Based on the estimated optimal location, some points were proposed for further expensive evaluations to improve the estimation. Mode-pursuing sampling model was also used to optimize MBDO Reference (Qiang & Changzhi (2014), Ramadan (2013), Simpson, Peplinski, Koch, Allen. (2001)). This model systematically generates more sample points around the mode of the black-box function and in the same time covers the entire sample space.

Constrained Optimization Problems, COP, were also solved in literatures (Cohon, Hedge, Martin, Richards (1987), Goldberg (1989), Hedar, & Fukushima (2002), Kazemi, Wang, Rahnamayan, & Gupta (2010), Moslem, Wang, Shahrayar, Kamal (2011), Sandgren (1990), Scholau, Welch, Jones (1998), Sharif,

Wang, El Mekkawy (2008)). Jones (2001) mentioned three general categorize for constraints handling methods; penalty-based methods, biasing-based methods, and multiobjective-based methods. Of the three categorize, penalty-based methods are the most used one in the literatures because it has a simple theory and straight forward implementation. The principle behind penalty-based methods is that any violation in the constraints must finally make the objective function value infeasible. This means that the constraints must be impeded in the objective function itself using an auxiliary function known as penalty function, hence, transforming the COP into unconstrained optimization problem UCOP. These penalty functions can be further categorized into interior and exterior penalty functions (Chelouah, Siarry (2005), Chen, Qiu, Jiao (2013)). The exterior penalty function is designed such that a sequence of infeasible points is generated to lead to the optimal solution for the COP (Kramer (2010)). This method utilizes the maximum function to incorporate the constraints into the objective function and thus makes the resulting UCOP a non smooth problem. The non smoothness in the resulting UCOP calls for using derivative-free-based optimization methods (Lam (2008)). However, the advantage of this method is that with a reasonably large penalty parameter, which is not required to approach infinity, it can yield the optimum solution of the COP from the UCOP.

Recently, hybrid stochastic-deterministic global optimization methods were gaining attention to solve COP (Coello, MezuraMontes (2002), Dellino, Kleijnen, Meloni (2010), Durand, Alliot (1999), Forrester, & Keane (2009), Huidae, Francisco, & Seth (2008)). These methods combine the benefits of the deterministic and stochastic methods. The deterministic methods such as Newton's method, conjugate gradient method and coordinate search method are known for their local search efficiency. The stochastic methods such as Genetic algorithm, Tabu search, and simulated annealing are known for their global search efficiency.

In this study, a free sampling algorithm and easyhybrid global optimization method is proposed to solve COP using the stochastic genetic algorithm method and the deterministic shrinking box method.

Genetic algorithm GA is rooted in Darwin's theory of evolution. The theory is based on natural selection process in which the strong individuals (those who fit well in their environment) survive and their offspring prosper while the weak individuals (those who do not fit well in their environment) die out and thus their offspring eventually vanish. GA is highly immune to be trapped in local optima as it has two operators to enhance the global search, namely, crossover and mutation operators (Coello, MezuraMontes (2002), Moghaddam, Wang, Yannou, Wu (2006), Wang, Cai, Zhou, Fan (2009), Wu, Chow (1995)). The purpose of using the shrinking box method is to narrow down the search space for the GA which will enhance its ability to find the optimal solution. The proposed hybrid algorithm consists of two loops an outer loop and an inner loop. The outer shrinking box loop utilizes the results obtained from the inner genetic algorithm loop to reduce the search space and then the outer loop feeds back the new reduced search space into the inner loop. This process continues until a stopping criterion is met.

The rest of the paper will be arranged as follows: section 2 introduces the proposed method. Section 3 describes the elements of the proposed method. Section 4 conducts ANOVA for the proposed method. Section 5 shows the experimentation. And finally, section 6 presents the conclusions.)

2. The Proposed Method

In this study a primary COP of the form:

$$\min_{\theta \in \Theta} f(\theta)$$

s.t.

$$g(\theta),$$

is considered where $f: \Theta \rightarrow \mathbb{R}$ is the objective function, Θ is a $\text{card}(\Theta)$ -dimensional sample space, θ is the set of the random variables, and $g(\theta)$ is the set of constraints.

Let $\hat{g}(\theta)$ be the complementary set of functions for $g(\theta)$ such that if $g(\theta)$ is satisfied, $\hat{g}(\theta)$ is not satisfied and vice versa. We can define a penalty function $\varphi(\hat{g}(\theta), \text{PenVal})$ as follows:

$$\varphi(\hat{g}(\theta), \text{PenVal}) = \sum_{i=1}^{NC} \text{PenVal}_i \times \max(0, \hat{g}_i(\theta)) \quad (1)$$

where PenVal_i is the penalty coefficient corresponding to variable i and it is sufficiently large value, NC is the total number of constraints. Then

$$\max(0, \hat{g}_i(\theta)) = \begin{cases} 0 & \text{if } g_i(\theta) \text{ is satisfied} \\ 1 & \text{if } g_i(\theta) \text{ is not satisfied} \end{cases} \quad (2)$$

This is true because $\hat{g}_i(\theta)$ and $g_i(\theta)$ are two mutually exclusive events due to the complementary relation between them.

Equation (1) can be used to penalize the primary objective function $f(\theta)$ if one or more values of the random variables θ did not satisfy one or more constraints in $g(\theta)$. Hence an auxiliary UCOP can be deduced from the primary COP as follows:

$$\Psi = \min[f(\theta) + \varphi(\hat{g}(\theta), \text{PenVal})] \quad (3)$$

It should be noticed here that the set of primary constraints $g(\theta)$ includes the box constraints and any other constraints like integer or binary constraints on the random variables θ . Hence the constraints $\theta \in \Theta$ are incorporated in $\hat{g}(\theta)$. Therefore these constraints will not show in the auxiliary UCOP.

Lam (2008) showed that the optimal solution and the optimal value of the UCOP are the same optimal solution and optimal value for the COP if the penalty values PenVal are infinite because

$$\lim_{\text{PenVal} \rightarrow \infty} \min[f(\theta) + \varphi(\hat{g}(\theta), \text{PenVal})] = \min_{\theta \in \Theta} f(\theta). \quad (4)$$

Unfortunately, making PenVal very large may cause serious computational difficulties (Lam (2008)). Reference (Jones, Schonlau, Welch (1998)) showed that PenVal does not need to be infinite for the UCOP to capture the optimal solution and the optimal value of the COP, it only needs to be sufficiently large.

As mentioned earlier, since the maximum function is integrated in the auxiliary UCOP, the auxiliary UCOP is not smooth. Therefore, all the derivative-based optimization methods cannot be applied directly to solve the UCOP. This matter calls for using derivative-free-based methods to solve it such as Genetic Algorithm.

In this study, a hybrid GA with Variance-based crossover VBC and Arithmetic-based mutation ABM operators along with a fitness function of the form Equation (3) will be used in the context of shrinking box method. This is a hybrid method between the stochastic nature of the GA and the deterministic nature of the shrinking box method. GA is used because it is good in global searching while shrinking box method is used to adaptively update and narrow the search space. This combination of stochastic-deterministic nature of the proposed method is aimed to increase the global and local search capabilities of the proposed method.

3. Elements of the Proposed Method

In this section we will present the different elements of the proposed method.

3.1 Shrinking Box

Let $\theta \in \Theta$ be the box constraints of the COP where Θ is the set of upper and lower limits of the random variables θ such that

$$\Theta = [l, u], \quad (5)$$

where the vectors l and $u \in \mathbb{R}^n$ represent the upper and lower bounds for θ respectively and $n = \text{card}(\theta)$.

The Shrinking box operator will adaptively change l and u according to the best solution found (best chromosome) as follows:

$$l_{i_{\text{new}}} = \left(\frac{l_{i_{\text{old}}} + g_{e_i}}{2} \right) \quad (6)$$

and

$$u_{i_{new}} = \left(\frac{u_{i_{old}} + ge_i}{2} \right), \tag{7}$$

where $l_{i_{new}}$, $l_{i_{old}}$ are the new lower limit and the old lower limit of the random variable i , $u_{i_{new}}$, $u_{i_{old}}$ are the new upper limit and the old upper limit of the random variable i , and ge_i is the gene value in the best chromosome corresponds to the random variable i .

3.2 Initial Population

The genes that constitute the initial population chromosomes will be generated randomly from \emptyset . The number of chromosomes in the initial population is equal to the population size PopSize which is related to the total generation number TGN as follows:

$$PopSize = \gamma \times TGN, \tag{8}$$

where γ is a parameter between 0 and 1 corresponding to population size proportion and it must be tuned. The initial population will be generated in two steps as follows:

While $j < PopSize + 1$

 For $i=1: n$

 Step 1

 Discretize the values between l_i and u_i into stp values, where stp is a discretization parameter that must be set through tuning.

 Give each discretized value a sequential number starting from 1 and ending at stp.

 Step 2

 Randomly select a value between 1 and stp and trace it back to its original value between l_i and u_i .

 Assign this value to ge_i .

 End

 Assign the vector ge to Chromosome j , i.e. Chr j .

 End while

3.3 Variance-Based Crossover

The set of chromosomes (population) at any generation can be represented as a matrix M in which its row numbers represent the chromosomes numbers and the column numbers represent the random variables (genes) numbers. This means that a general gene can be represented as ge_{ij} where j stands for chromosome j and i stands for gene i , i.e. variable i . Hence $Chri_j$ stands for gene i in chromosome j . The genotype representation of the population is as follows:

g_{11}	g_{12}	g_{13}	-	-	-	g_{1n}	-	-	-	g_{1n}
g_{21}	g_{22}	g_{23}	-	-	-	g_{2n}	-	-	-	g_{2n}
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
g_{j1}	g_{j2}	g_{j3}	-	-	-	g_{jn}	-	-	-	g_{jn}
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
$g_{PopSize}$	$g_{PopSize}$	$g_{PopSize}$	-	-	-	$g_{PopSize}$	-	-	-	$g_{PopSize}$

Figure 1. Genotype representation for the population

The VBC depends on the variance for the variable values in the different chromosomes. The steps involved in VBC calculations are as follows:

Step 1

Assign a zero-one score to the values of the variables as follows:

$$Sc_{ge_{ij}} = \frac{ge_{ij} - \min_j(ge_{ij})}{\max_j(ge_{ij}) - \min_j(ge_{ij})} \quad \forall i = 1, 2, 3, \dots, n \quad \forall j = 1, 2, 3, \dots, \text{PopSize} \quad (9)$$

Step 2

Calculate the dispersion in the $Sc_{ge_{ij}}$ values for the each variable (gene) over the population using the variance formula as follows:

$$\text{Var}(\theta_i) = \frac{\sum_{j=1}^{\text{PopSize}} (Sc_{ge_{ij}} - \bar{Sc}_{ge_{ij}})^2}{\text{PopSize} - 1}, \quad \forall i = 1, 2, 3, \dots, n \quad (10)$$

where $\bar{Sc}_{ge_{ij}}$ the average of variable i values over the population and is calculated as:

$$\bar{Sc}_{ge_{ij}} = \frac{\sum_{j=1}^{\text{PopSize}} Sc_{ge_{ij}}}{\text{PopSize}} \quad (11)$$

Step 3

Find the indices for the $(n/2)$ genes corresponding to the smallest $(n/2)$ variances (LowVar) and assign the remaining indices to HighVar

Step 4

Generate NOffSpr offspring from the best chromosome found in the generation (BGChr) where NOffSpr is the number of offspring generated in the generation. The generation of the offspring OffSpr is as follows:

$$\begin{aligned} \text{OffSpr}_{\text{LowVar},j} &= \text{BGChr}_{\text{LowVar},j} \\ \text{OffSpr}_{\text{HighVar},j} &= \text{Chr}_{\text{HighVar},j} \end{aligned}$$

The number of offspring generated in any generation is changing with the generation number as follows:

$$\text{NOffSpr} = \text{IniOffSpr} + \text{GN} \quad (12)$$

where GN is the generation number, IniOffSpr is initial number of offspring and it is a parameter that must be tuned. This dynamic change in NOffSpr is aimed to reduce the variability toward the end of the generations to allow for a delicate search while keeping the variability high at the beginning to explore the search space adequately.

3.4 Arithmetic-Based Mutation

The ABM consists of the following 3 steps:

Step 1

Choose $\text{MuGe} = \text{floor}(r_m \times n)$ random numbers between 1 and n , where r_m is the mutation rate and it is taken to be 0.1.

Choose a $\text{MuOff} = \text{floor}(r_m \times \text{PopSize})$ random numbers between 1 and PopSize .

Step 2

Choose $\kappa = \text{floor}(r_m \times \text{PopSize})$ random numbers between 0 and 1

Step 3

Calculate the mutated offspring as follows:

$$\text{OffSpr}(\text{MuGe}, \text{MuOff}) = \kappa \times \text{OffSpr}(\text{MuGe}, \text{MuOff}) + (1 - \kappa) \times \text{BGChr}(\text{MuGe}, \text{MuOff})$$

3.5 Immigration

The aim of the immigration operator is to keep the population diverse (Ramadan (2013)) and thus mitigate the effect of premature convergence. The immigration operator can be performed as follows: generate $(\text{PopSize} - \text{NOffSpr} - 1)$ chromosomes as explained in section 2.2.2. The number of immigrants reduces as the generation number increases because the offspring number increases according to Equation (4). This dynamic change in immigrant numbers aims to increase the variability in the population at the beginning to increase the exploration power of the method and mitigate the premature convergence problem. In addition, it aims to decrease the

variability toward the end of the generations.

At this point, the number of chromosomes constituting the existing population is PopSize-1. The final step in the population preparation is the step of including the best chromosome found in the previous generation BGChr in the existing generation. This can be done by demanding that the chromosome number PopSize be the same as BGChr. This will make the population size equals to PopSize.

3.6 Stopping Criteria

The stopping criterion for this proposed method is to stop the algorithm when the objective function enhancement in the outer loop is zero while the stopping criterion for the GA i.e., the inner loop is when the generation number reaches TGN generations.

The flow diagram representing the proposed method is given in Figure 2 as follows:

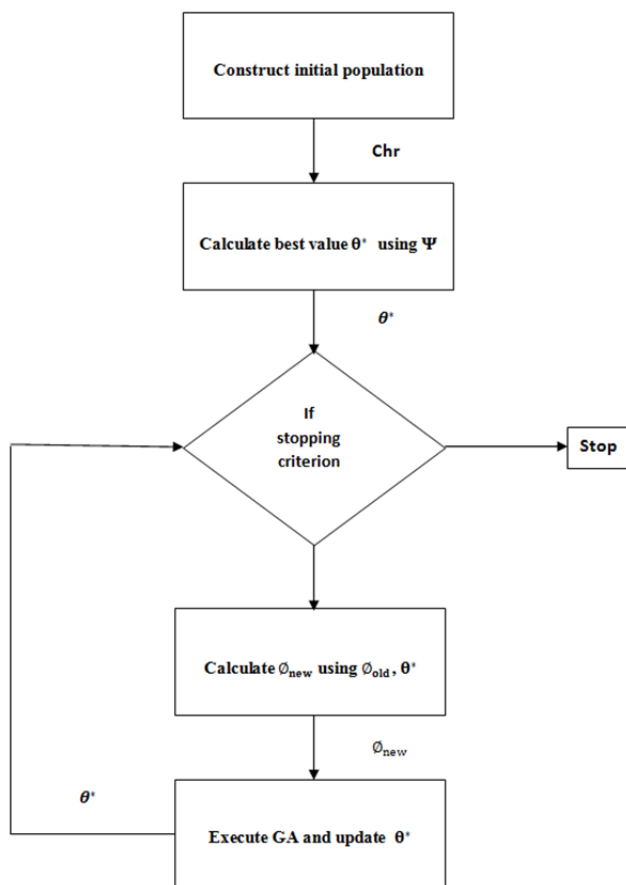


Figure 2. Flow diagram for SBPGA

4. Analysis of Variance Study for the Proposed Method

In the proposed SBPGA, the parameters that need tuning is 3, namely: stp, γ , and IniOffSpr. ANOVA study was conducted on the proposed method to assess the importance of the 3 parameters on its performance. Three levels were taken for each parameter and the average of 100 replications for the fitness function value of the six-hump camel-back SC benchmark problem was recorded as the response variable. The expression of the SC is giving below:

$$f_{SC} = 4x_1^2 + \frac{21}{10}x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

s.t

$$x_1, x_2 \in [-2, 2]$$

The ANOVA results are shown in Table 1.

Table 1. ANOVA results for the parameters of the proposed method

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Discretization step stp	0.0003	2	0.00015	43.79	0
Population size proportion γ	0.0001	5	2 0.00007	22.03	0.0006
Initial size of offspring, IniOffSpr	0.0000	3	2 0.00001	4.13	0.0585
stp * γ	0.0001	1	4 0.00003	8.17	0.0063
stp * IniOffSpr	0.0000	3	4 0.00001	2.07	0.1772
γ * IniOffSpr	0.0000	3	4 0.00001	2.01	0.1854

Column 1 shows the source of variability and column 6 shows the p-value for the F statistic. The test is conducted at 5% significance level. The results table shows that the discretization step size stp and the population size proportion γ significantly affect the best objective function value found. These results are interesting as it says that the discretization step size and the population size proportion are the only two parameters among the three parameters that significantly affect the performance of the model. The table also shows that the interaction between the stp and γ is significant. This result is important as it suggests that discretization step size and population size proportion must be tuned simultaneously as they both affect each other and affect the best value found. On the other hand, the results show that the initial size of offspring IniOffSpr and both of its interactions are not significant.”

4. Experimentation

The proposed method has been tested using 11 benchmark problems. The formulations of the problems are described in this paper for convenience. Each problem was solved 100 times and the average of these replications was used to construct a 95% confidence interval on the mean solution along with other central tendency and dispersion measures. A comparative analysis is also shown that compares the results of the proposed method with the results obtained from literatures. Moreover, the success rate for the different problems was calculated according to [44] and used as a performance measure for the proposed method. The success rate has the form:

$$\text{Success rate} = \frac{\text{successful implementation}}{\text{total implementation}},$$

where

$$\text{Successful implementation} = \frac{f^* - f^{**}}{|f^{**}| + 1},$$

and f^* and f^{**} is the obtained best solution and the best known solution, respectively. The machine used to solve the problems has the following specifications: Manufacturer HP, Model HPE-500f, Processor AMD phenon (tn) IIX6 1045T processor 2.70GHz, RAM 8.0 GB, system 64-bit operating system.

The descriptions and the results of the problems are as follows:

A simple QF function with two variables. Reference [28] reported the mathematical expression for this function is as follows:

$$f_{\text{QF}} = (x_1 + 1)^2 + (x_2 - 1)^2,$$

s.t

$$x_1, x_2 \in [-3, 3]$$

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was [0.981, 4.834]E-5 while the best value found was 3.8964e-12 corresponding to the best solution of [-1.000, 1000]. The variance of 9.66E-9 indicates that the dispersion in the values of the best values is very low and thus the method was robust in this problem. The average time needed for each replication was 3.47s which is reasonable. Hence the method was efficient in terms of computational time in this problem. Table 2 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [28].

Table 2. Comparison between SBPGA and other literatures for QF problem

$$f^{**} = 0$$

Reference	f^*
SBPGA	3.8964E-12
Reference [28]	0

Six-hump camel-back function (SC) with two variables. Refs.[28, 29, 36] among other references used this problem, the mathematical expression for this function is as follows:

$$f_{SC} = 4x_1^2 + \frac{21}{10}x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

s.t

$$x_1, x_2 \in [-2, 2]$$

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was [-1.0315 -1.0315] while the best value found was -1.0315 corresponding to the best solution of [0.0899, -0.7127]. The variance of 2.3295E-08 indicates that the dispersion in the values of the best values is very low and thus the method was robust in this problem. The average time needed for each replication was 3.51s which is reasonable. Hence the method was efficient in terms of computational time in this problem. Table 3 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [28].

Table 3. Comparison between SBPGA and other literatures for SC problem

$$f^{**} = -1.032$$

Reference	f^*
SBPGA	-1.0315
Reference [28]	-1.030

Goldstein–Price function (GP) with two variables. Refs. [28, 37] among other references used this problem, the mathematical expression for this problem as follows:

$$f_{GP} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)],$$

s.t

$$x_1, x_2 \in [0, 1]$$

The rate of success for this problem was 81%. The 95% confidence interval on the mean of the best value was [3.0144, 3.0383] while the best value found was 3.0264 corresponding to the best solution of [0, -1]. The variance of 0.3541 indicates that the dispersion in the values of the best values is somewhat high and thus the method was not robust in this problem. The average time needed for each replication was 3.65s which is reasonable. Hence the method was efficient in terms of computational time in this problem. Table 4 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [28].

Table4. Comparison between SBPGA and other literatures for GPr problem

$$f^{**} = 3$$

Reference	f^*
SBPGA	3
Reference [28]	3.005

Hartman Function (HF) with 6 variables. Refs. [28, 38] among other references used this problem, the mathematical expression for this function as follows:

$$f_{HF} = - \sum_{i=1}^4 c_i \exp \left[- \sum_{j=1}^n \alpha_{ij} (x_j - p_{ij})^2 \right],$$

s.t

$$x_i \in [0,1]$$

where

i	$\alpha_{ij}, j = 1, 2, \dots, 6$						c_i
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2

And

i	$p_{ij}, j = 1, 2, \dots, 6$					
1	0.1696	0.5569	0.0124	0.8283	0.5886	0.1696
2	0.4135	0.8307	0.3736	0.1004	0.9991	0.4135
3	0.1451	0.3522	0.2883	0.3047	0.6650	0.1451
4	0.8828	0.8732	0.5743	0.1091	0.0381	0.8828

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was [-3.2778, -3.2538] while the best value found was -3.3224 corresponding to the best solution of [0.20170.15000.47690.27530.31170.6573]. The variance of 0.0037 indicates that the dispersion in the values of the best values is somewhat low and thus the method was reasonably robust in this problem. The average time needed for each replication was 47.13s which less than a minute and thus it is considered reasonable. Hence the method was fairly efficient in terms of computational time in this problem. Table 5 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [28].

Table 5. Comparison between SBPGA and other literatures for HF problem

$f^{**} = -3.322$	
Reference	f^*
SBPGA	-3.322
Reference [28]	-3.322

Griewank function (GN) with two variables. Refs. [28, 39] among other references used this problem, the mathematical expression for this function as follows:

$$f_{GN} = \sum_{i=1}^2 \frac{x_i^2}{200} - \prod_{i=1}^2 \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1,$$

s.t

$$x_i \in [-100,100]$$

The rate of success for this problem was 98%. The 95% confidence interval on the mean of the best value was $[-1.531, 9.604]E-4$ while the best value found was 0 corresponding to the best solution of $[0, 0]$. The variance of $8.0687E-06$ indicates that the dispersion in the values of the best values is low and thus the method was robust in this problem. The average time needed for each replication was 3.40s which is low. Hence the method was efficient in terms of computational time in this problem. Table 6 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [28].

Table6. Comparison between SBPGA and other literatures for GN problem

$f^{**} = 0$	
Reference	f^*
SBPGA	0
Reference [28]	0

Design of a Pressure Vessel (PV) with 4 variables and 3 constraints. This problem was solved in many references including [28, 29, 35, 40]. The pressure vessel is shown in Figure 3 with the corresponding dimensions. The shell consists of two welded halves of rolled carbon steel ASME SA 203 grade B steel plates. There are 4 design variables: radius (x_1), spherical head thickness (x_2), Shell thickness (x_3), and length (x_4).

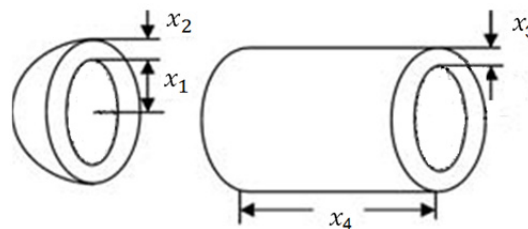


Figure 3. A pressure Vessel [28]

The objective is to minimize the total cost given by:

$$\min Z_{GN} = 0.6224x_3x_1x_4 + 1.7781x_2x_1^2 + x_3^2(3.1661x_4 + 19.84x_1)$$

Subject to ASME boiler and pressure code for wall thickness x_3 and x_2 constraints given by:

$$g_1 = x_3 - 0.0193x_1 \geq 0 ,$$

$$g_2 = x_2 - 0.00954x_1 \geq 0 ,$$

and tank volume

$$g_3 = \pi x_1^2 x_4 + \frac{3}{4} \pi x_1^3 - 1.296E6 \geq 0 .$$

The radius and the length are positive integers, while the shell thickness and the spherical head thickness are positive real numbers. The ranges for the design variables are as follows:

$$x_1 \in [25, 150], x_2 \in [0.625, 1.0], x_3 \in [1.0, 1.375], \text{ and } x_4 \in [25, 240]$$

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was $[7012.77014, 4]$ while the best value found was 7007.3 corresponding to the best solution of $[51.82290, 62501.000284, 5169]$. The variance of $1.9182e-05$ indicates that the dispersion in the values of the best values is low and thus the method was robust in this problem. The average time needed for each replication was 62.9s which is a little bit higher than a minute. Hence the method was fairly efficient in terms of computational time in this problem. Table 7 shows a comparison between the results obtained for this problem using SBPGA

and the results obtained using Refs. [46, 48].

Table 7. Comparison between SBPGA and other literatures for PV problem

$$f^{**} = 7006.8$$

Reference	f*
SBPGA	7007.3
GA Reference [46]	7207.494
IHS Reference [48]	7197.730

Gear Train Problem (GT) with 4 variables and integer variables constraints. This problem was solved in Reference [29]. The objective of this design problem is to find the closest gear ratio to the standard parameter given by the designer in [41] which is 0.14428. There are 4 design variables corresponding to the numbers of teeth of the four gears in the setup as shown in Figure 4.

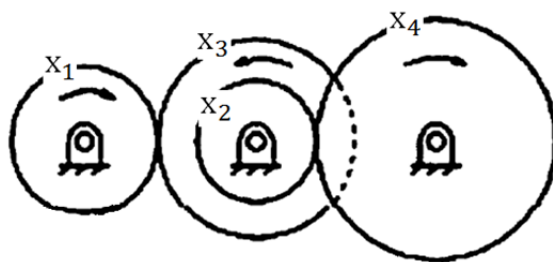


Figure 4. Train Gear Problem [29]

The objective is to minimize difference between the teeth ratios and the standard parameter as follows:

$$\min Z_{GT} = \left(0.14428 - \frac{x_1 x_2}{x_3 x_4}\right)^2$$

s.t

$$x_i \in [12, 60] \forall i \in \{1, 2, 3, 4\}$$

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was [5.5988.708]E-10 while the best value found was 2.7009E-12 corresponding to the best solution of [19164943]. The variance of 6.2916e-19 indicates that the dispersion in the values of the best values is low and thus the method was robust in this problem. The average time needed for each replication was 6.3s which is reasonable low. Hence the method was reasonably efficient in terms of computational time in this problem. Table 8 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [47].

Table 8. Comparison between SBPGA and other literatures for GT problem

$$f^{**} = 2.7E - 12$$

Reference	f*
SBPGA	2.7E-12
MIBB-SQP Reference [47]	5.7E-6
MIHDE Reference [47]	2.7E-12

Weight of a spring problem (WS) with 3 variables and 3 constraints. This problem was solved in many references including Refs. [35, 42]. The objective of this design problem is to minimize the weight of the spring.

The design variables are; wire diameter x_1 , the mean coil diameter x_2 , and the number of active coils x_3 . Figure 5 shows a schematic diagram for the coil spring.

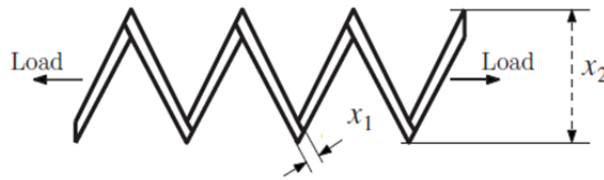


Figure 5. Schematic diagram for WS problem adapted from [35]

The model is given as:

$$\min Z_{WS} = (x_3 + 2)x_2x_1^2$$

Subject to:

$$g_1 = \frac{x_2^3x_3}{71875x_1^4} \geq 1$$

$$g_2 = \frac{x_2(4x_2 - x_1)}{12566x_1^3(x_2 - x_1)} + \frac{2.46}{12566x_1^2} \leq 1$$

$$g_3 = \frac{140.54x_1}{x_2^2x_3} \geq 1$$

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was [0.01268 , 0.012716] while the best value found was 0.012626 corresponding to the best solution of [0.05170.357311.2284]. The variance of 6.4127E-09 indicates that the dispersion in the values of the best values is low and thus the method was robust in this problem. The average time needed for each replication was 28.4s which is reasonable. Hence the method was fairly efficient in terms of computational time in this problem. Table 9 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Refs. [48, 49].

Table 9. Comparison between SBPGA and other literatures for WS problem

$f^{**} = N/A$	
Reference	f^*
SBPGA	0.012626
GA Reference [48]	0.012681
IHS Reference [49]	0.0126706

Ackely Function (A_y) with 10 variables: Reference [44] used this function, the mathematical formulation for this function is as follows:

$$f(x) = 22.7183 - 20 \times \exp\left(-0.2 \sqrt{\frac{1}{10} \sum_{i=1}^{10} x_i^2}\right) - \exp\left(\frac{1}{10} \sqrt{\sum_{i=1}^{10} \cos cx_i}\right),$$

s.t

$$x_i \in [-32 \ 32], \forall i = 1, 2, 3, \dots, 10$$

The rate of success for this problem was 44%. The 95% confidence interval on the mean of the best value was [0.00560.3646] while the best value found was 2.1620E-11 corresponding to the best solution of [0.2306-0.02550.20740.11610.34160.31360.30940.01590.0161-0.3845]E-5. The variance of 0.8387 indicates that the dispersion in the values of the best values is extremely high and thus the method failed in this problem. The average time needed for each replication was 167.5s which is high. Hence the method failed also in terms of computational time in this problem. Table 10 shows a comparison between the results obtained for this problem

using SBPGA and the results obtained using Reference [44].

Table 10. Comparison between SBPGA and other literatures for Ay problem

$f^{**} = 0$	
Reference	f^*
SBPGA	2.1620E-11
GA Reference [44]	3.6309e-4
GAHJ [44]	1.6083e-5

Even though SBPGA failed in terms of robustness and efficiency, it was able to find a solution much better than the other two methods reported in [44].

Beale Function (Be) with 2 variables: Ref [44] used this function, the mathematical formulation is as follows:

$$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2,$$

s.t

$$x_i \in [-4.5 \ 4.5], \forall i = 1, 2$$

The rate of success for this problem was 100%. The 95% confidence interval on the mean of the best value was [0.06230.4474]E-3 while the best value found was 3.6732E-24 corresponding to the best solution of [3.00000.5000]. The variance of 9.6504E-07 indicates that the dispersion in the values of the best values is low and thus the method was robust in this problem. The average time needed for each replication was 30.4s which is reasonable. Hence the method was fairly efficient in terms of computational time in this problem. Table 11 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [44].

Table 11. Comparison between SBPGA and other literatures for Be problem

$f^{**} = 0$	
Reference	f^*
SBPGA	3.6732E-24
GA Reference [44]	4.9422e-10
HJM Reference [44]	1.7672e-12
GAHJ [44]	2.0463e-12

Zakharov function (ZV) with 10 variables: Ref [44] used this function, the mathematical formulation is as follows:

$$f(x) = \sum_{i=1}^{10} x_i^2 + \left(\sum_{i=1}^{10} 0.5 \times i \times x_i \right)^2 + \left(\sum_{i=1}^{10} 0.5 \times i \times x_i \right)^4$$

s.t

$$x_i \in [-5 \ 10], \forall i = 1, 2, 3, \dots, 10$$

The rate of success for this problem was 79%. The 95% confidence interval on the mean of the best value was [0.00350.0972] while the best value found was 2.4068E-07 corresponding to the best solution of [-0.20310.24420.15150.11640.07780.0407-0.2179-0.1487-0.06320.1475]E-3. The variance of 0.0570 indicates that the dispersion in the values of the best values is not low and thus the method was not robust in this problem. The average time needed for each replication was 166.6s which is high. Hence the method was not efficient in terms of computational time in this problem. Table 12 shows a comparison between the results obtained for this problem using SBPGA and the results obtained using Reference [44].

Table 12. Comparison between SBPGA and other literatures for ZV problem

$f^{**} = 0$	
Reference	f^*
SBPGA	2.4068E-07
GA Reference [44]	4.7725E-5
HJM Reference [44]	1.5609E-5
GAHJ [44]	6.1889E-8

5. Conclusions

This paper, presents a simple, fast and forward global optimization method for COP based on shrinking box and GA methods. The proposed method showed a good degree of robustness in 8 out of the 11 problems, hence the method is considered fairly robust. In addition, the proposed method showed a good degree of computational time efficiency as it was efficient in 9 out of the 11 problems. Furthermore, the proposed method was able to have 100% success rate in 7 out of the 11 problems. The success rates for the remaining 4 problems are as follows: the GN problem has 98% success rate which is very high. Problems GP and ZV have 81% and 79% success rates respectively which are reasonable. Unfortunately, Ay problem has a low success rate of 44%. This low success rate is a result of the dimensionality problem.

Even though there is no guarantee that the best values found by this method are feasible due to using the auxiliary UCOP instead of the primary COP, all the best solutions found in the 11 problems are feasible. This is due to using high penalty value in the auxiliary function the matter that helped in eliminating the infeasible solutions. Unfortunately the high penalty value reduced the computational efficiency of the proposed method.

Based on these results, we believe that the proposed method is fairly robust and efficient global optimization method for COP whether they are continuous or discrete even though it has a stochastic element in it. Moreover, the proposed method is a standalone method that does not need any other algorithms such as sampling algorithms. Like most of the evolutionary algorithms, this method suffers from the curse of dimensionality and premature convergence problems. The shrinking box strategy was used in the proposed method to mitigate the effect of these problems.

Acknowledgments

The author is grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research (Grant No.DRGS-2015-2016).

References

- Abu Bakar, S., Ramlan, M., & Muhammad, S.(2007). Reducing Premature Convergence Problem through Numbers Structuring in Genetic Algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 7(4), 215-217.
- Bazaraa, M., Sherali, D., & Shetty, C.(2006). *Nonlinear Programming: Theory and Algorithm (Third Edition)*. Wiley Online Library. <http://dx.doi.org/10.1002/0471787779>
- Belegundu, D.,& Arora, S. (1985). A Study of Mathematical Programming Methods for Structural Optimization. Part II: Numerical Results. *Int. J. Numer. Methods Eng.*, 21(9), 1601–1623. <http://dx.doi.org/10.1002/nme.1620210905>
- Branin, H., & Hoo, K. (1972). A Method for Finding Multiple Extrema of a Function of N Variables,” *Numerical Methods for Non-Linear Optimization.F. Lootsma, Academic*, 231–237.
- Camp, G. (1955). Inequality-constrained stationary-value problems. *Journal of the Operations Re-search Society of America*, 3, 548-550. <http://dx.doi.org/10.1287/opre.3.4.548a>
- Carroll, C., & Fiacco, A.(1961). The created response surface technique for optimizing nonlinear restrained systems. *Operations Research*, 9, 169-185. <http://dx.doi.org/10.1287/opre.9.2.169>
- Chelouah, R., & Siarry, P.(2005). A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multim minima functions. *European Journal of Operational Research*, 161, 636-654. <http://dx.doi.org/10.1016/j.ejor.2003.08.053>
- Chen, Z., Qiu, S.,& Jiao, Y. (2013).A penalty-free method for equality constrained optimization. *Journal of*

- Industrial and Management Optimization*, 9, 391-409. <http://dx.doi.org/10.3934/jimo.2013.9.827>
- Coello, C. A. C., & MezuraMontes, E. (2002). Constraint-Handling in Genetic Algorithms Through the Use of Dominance-Based Tournament Selection. *Adv. Eng. Inf.*, 16(3), 193-203. [http://dx.doi.org/10.1016/S1474-0346\(02\)00011-3](http://dx.doi.org/10.1016/S1474-0346(02)00011-3)
- Cohon, P., Hedge, S., Martin, W., & Richards, D.(1987). *Punctuated equilibria: A parallel genetic algorithm*. Proceedings of the Third International Conference on Genetic Algorithms and Their Applications.
- Dellino, G., Kleijnen, C., & Meloni, C.(2010). Robust optimization in simulation: Taguchi and response surface methodology. *Int. J. Prod. Econ.*, 125(1), 52-59. <http://dx.doi.org/10.1016/j.ijpe.2009.12.003>
- Durand, N., & Alliot, J.(1999). *A combined nelder-mead simplex and genetic algorithm*. In Proceed-ings of the Genetic and Evolutionary Computation Conference GECCO, 99, 1-7.
- Forrester, J.,& Keane, J. (2009). Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.*, 45(1-3), 50-79. <http://dx.doi.org/10.1016/j.paerosci.2008.11.001>
- Goldberg, D. (1989). Genetic algorithms in search, optimization and machine learning.
- Hedar, A., & Fukushima, M.(2002). *Hybrid simulated annealing and direct search method for nonlinear global optimization*. Department of Applied Mathematics & Physics Kyoto University, 17, 891-912.<http://dx.doi.org/10.1080/1055678021000030084>
- Huidae, C., Francisco, O., & Seth, D. (2008). A derivation of the number of minima of the Griewank function. *Applied Mathematics and Computation*, 204(2), 694-701. <http://dx.doi.org/10.1016/j.amc.2008.07.009>
- Jones, R. (2001). A taxonomy of global optimization methods based on response surfaces. *J. Glob. Optim.*, 21(4), 345-383. <http://dx.doi.org/10.1023/A:1012771025575>
- Jones, R., Schonlau, M., & Welch, J. (1998). Efficient global optimization of expensive black-box functions. *J. Glob. Optim.*, 13(4), 455-492. <http://dx.doi.org/10.1023/a:1008306431147>
- Kazemi, M., Wang, G., Rahnamayan, S., & Gupta, K.(2010).*Constraint Importance Mode Pursuing Sampling for Continuous Global Optimization*. ASME Paper.
- Kramer, O.(2010). A review of constraint-handling techniques for evolution strategies. *Applied Computational Intelligence and Soft Computing*. <http://dx.doi.org/10.1155/2010/185063>
- Lam, Q.(2008). *Sequential adaptive designs in computer experiments for response surface model fit*. Ph.D. thesis, The Ohio State University.
- Mahdavi, M., Fesanghary, M., & Damangir, E.(2007). An Improved Harmony Search Algorithm for Solving Optimization Problems.*Appl. Math. Comput.*, 188(2),1567-1579.
- Mallipeddi, R., & Suganthan, N.(2010). Ensemble of constraint handling techniques. *Evolutionary Computation, IEEE Transactions on*, 14, 561-579.
- Moghaddam, R., Wang, G., Yannou, B., & Wu, C. (2006).*Applying Constraint Programming for Design Space Reduction in Metamodeling Based Optimization*. 16th International Institution for Production Engineering Research_CIRP_International Design Seminar.
- Moslem, K., Wang, G., Shahrayar, R., & Kamal, G. (2011). Metamodel-Based Optimization for Problems with Expensive Objective and Constraint Functions. *Journal of Mechanical design*,133.
- Picheny, V., Wagner, T., & Ginsbourger, D.(2012). A benchmark of kriging-based infill criteria for noisy optimization Functions.
- Pierskalla, W.(1968). Mathematical programming with increasing constraint functions.*Management Science*, 15, 416-425.
- Wilde, D.(1978). A hybrid method combining genetic algorithm and hooke-jeeves method For constrained global optimization. *Journal of industrial and management optimization*, 10(4), 1279-1296. <http://dx.doi.org/10.3934/jimo.2014.10.1279>
- Ramadan, S. (2013). Reducing Premature Convergence Problem in Genetic Algorithm:Application on Travel Salesman Problem. *Computer and Information Science*, 6(1).
- Ramadan, S.,& Ramadan, K. (2012). Hybrid Two-Stage Algorithm for Solving Transportation Problem. *Modern Applied Science*, 6(4).
- Regis, R., & Shoemaker, C.(2005). Constrained Global Optimization of Expensive Black Box Functions Using

- Radial Basis Functions. *J. Global Optim.*, 31(1), 153–171. <http://dx.doi.org/10.1007/s10898-004-0570-0>
- Sandgren, E. (1990). Nonlinear Integer and Discrete Programming in Mechanical Design Optimization. *ASME J. Mech. Des.*, 112(2), 223–229. <http://dx.doi.org/10.1115/1.2912596>
- Scholau, M., Welch, J., & Jones, D. (1998). Global versus local search in constrained optimization of computer models. In N. Flournoy, W. F. Rosenberger, & W. K. Wong (Eds.), *New Development and Applications in Experimental Design* (pp. 11–25), Institute of Mathematical Statistics.
- Sharif, B., Wang, G., & El Mekkawy, T. (2008). Mode Pursuing Sampling Method for Discrete Variable Optimization on Expensive Black-Box Functions. *ASME J. Mech. Des.*, 130(2). <http://dx.doi.org/10.1115/1.2803251>
- Simpson, T., Peplinski, J., Koch, P., & Allen, J. (2001). Metamodels for Computer-Based Engineering Design: Survey and Recommendations. *Eng. Comput.*, 17(2), 129–150. <http://dx.doi.org/10.1007/pl00007198>
- Torn, A., & Zilinskas, A. (1989). *Global Optimization*. Springer-Verlag New York, Inc. New York, NY, USA. <http://dx.doi.org/10.1007/3-540-50871-6>
- Ursem, K. (2002). Diversity-guided Evolutionary Algorithms. *Lecture Notes in Computer Science*, 2439, 462–471. http://dx.doi.org/10.1007/3-540-45712-7_45
- Vasconcelos, J., Ramirez, J., Takahashi, R., & Saldanha, R. (2001). Improvements in genetic algorithms. *Magnetics, IEEE Transactions on*, 37, 3414–3417.
- Wang, G., & Shan, S. (2007). Review of Metamodeling Techniques in Support of Engineering Design Optimization. *ASME J. Mech. Des.*, 129(4), 370–380. <http://dx.doi.org/10.1115/1.2429697>
- Wang, G., & Simpson, W. (2002). *Fuzzy clustering based hierarchical metamodeling for design optimization*. 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia. <http://dx.doi.org/10.2514/6.2002-5574>
- Wang, G., Dong, Z., & Aitchison, P. (2001). Adaptive response surface method—a global optimization scheme for computation-intensive design problems. *Journal of Engineering Optimization*, 33(6), 707–734. <http://dx.doi.org/10.1080/03052150108940940>
- Wang, L., Shan, S., & Wang, G. (2004). Mode-Pursuing Sampling Method for Global Optimization on Expensive Black-Box Functions. *Eng. Optimiz.*, 36(4), 419–438. <http://dx.doi.org/10.1080/03052150410001686486>
- Wang, Y., Cai, Z., Zhou, Y., & Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization*, 37, 395–413. <http://dx.doi.org/10.1007/s00158-008-0238-3>
- Wilde, D. (1978). *Globally Optimal Design*. Wiley, New York.
- Wu, S., & Chow, P. (1995). Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems via Meta-Genetic Parameter Optimization. *Eng. Optimiz.*, 24(2), 137–159. <http://dx.doi.org/10.1080/03052159508941187>
- Yannou, B., & Harmel, G. (2006). Use of Constraint Programming for Design. *Advances in Design*, 145–157.
- Yannou, B., Moreno, F., Thevenot, H., & Simpson, T. (2005). *Faster Generation of Feasible Design Points*. ASME Paper.
- Yu, C., Teo, L., Zhang, L., & Bai, Y. (2010). A new exact penalty function method for continuous inequality constrained optimization problems. *Journal of Industrial and Management Optimization*, 6, 895. <http://dx.doi.org/10.3934/jimo.2010.6.895>
- Zhi, Q., Meng, H., & Dang, C. (2009). A penalty function algorithm with objective parameters for nonlinear mathematical programming. *Journal of Industrial and Management Optimization*, 5, 585–601. <http://dx.doi.org/10.3934/jimo.2009.5.585>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).