Hindawi
Advances in Civil Engineering
Volume 2019, Article ID 7475156, 9 pages
https://doi.org/10.1155/2019/7475156

Research Article

Finding Plausible Optimal Solutions in Engineering Problems Using an Adaptive Genetic Algorithm

Muslum Kilinc[1] **and Juan M. Caicedo**[2]

[1] *Erciyes University, Department of Civil Engineering, Kayseri 38039, Turkey*
[2] *University of South Carolina, Department of Civil and Environmental Engineering, Columbia, SC 29208, USA*

Correspondence should be addressed to Muslum Kilinc; kilinc@erciyes.edu.tr

Received 23 October 2018; Revised 9 January 2019; Accepted 13 January 2019; Published 27 February 2019

Academic Editor: Flavio Stochino

In engineering, optimization applications are commonly used to solve various problems. As widely known, solution of an engineering problem does not have a unique result; moreover, the solution of a unique problem may totally differ from one engineer to another. On the other hand, one of the most commonly used engineering optimization methods is genetic algorithm that leads us to only one global optimum. As to mention, engineering problems can conclude in different results from the point of different engineers' views. In this study, a modified genetic algorithm named multi-solution genetic algorithm (MsGA) based on clustering and section approaches is presented to identify alternative solutions for an engineering problem. MsGA can identify local optima points along with global optimum and can find numerous solution alternatives. The reliability of MsGA was tested by using a Gaussian and trigonometric function. After testing, MsGA was applied to a truss optimization problem as an example of an engineering optimization problem. The result obtained shows that MsGA is successful at finding multiple plausible solutions to an engineering optima problem.

1. Introduction

It is known that optimization method is very important for engineering fields to obtain appropriate or economic solutions of an engineering application [1] For an engineering application example, optimization methods are used in determining the minimum weight when designing an aircraft and aerospace structures. Optimal trajectory of a rocket or aircraft can be determined with the aid of optimization. In the field of structural engineering, optimization can be a useful tool in the design of civil engineering structures such as frame domes, roofs, bridges, electrical transmission lines, antenna towers, roller coasters, tower cranes etc., to calculate the minimum cost.

An extended summary, discussion, and comparison on the optimization of metaheuristic method was made by Dogan and Olmez [2] studied on vortex search algorithm. Different metaheuristic algorithms are generally used in optimization of real-life engineering application. As an example, in one of the recent studies, a newly developed metaheuristic approach is used to optimize 25-bar, 72-bar, and 200-bar structures and a 26-story space tower [3]. Among the metaheuristic algorithms, for example, a swarm-based metaheuristic algorithm called Emperor Penguin Optimizer is developed to solve a 25-bar truss problem and six other engineering design problems [4]. Briefly, metaheuristic algorithms search in a certain solution domain in order to obtain the best optimal result.

Metaheuristic algorithms have generally two different solution approaches; one of them is single-solution approach and the other is multiple-solution approach. But even the multiple-solution approach method does not have the capability of capturing local optimal solution alternatives. Therefore, a new method is needed to find local optimal alternatives in the solution of this kind of problems in engineering optimization.

Metaheuristic algorithms can be divided into 4 main groups. Evolutionary algorithms like genetic algorithms

(GA), physics-based algorithms like simulated annealing (SA), swarm-based algorithms like particle swarm optimization (PSO), and nature-inspired algorithms like firefly algorithm (FA). Moreover, traditional genetic algorithm (TGA) is thought to be the best approach among these methods to find a unique optimal solution in complex search space for different common engineering problems [5].

The evolutionary algorithms which are inspired from natural selection of the biological evolution process; genetic algorithm (GA) is a popular method prevalently used. In detail, genetic algorithm (GA) is inspired by the mutation, reproduction, crossover, and selection abilities of the biological evolution process. Each algorithm uses an objective function to optimize the problem. TGA is achieved by encoding the characteristics of the optimization problem in an individual "chromosome" or "genome" which mimics the evolutionary process. TGAs can use binary or real genes depending on the type of problem considered. Individuals in a binary TGA are described by vectors of ones and zero. In a real parameter TGA, the individuals are described by a vector of real numbers. Genetic algorithms for global optimization problems have been used in many structural engineering applications such as discrete optimization of structures [6], sizing, shape, and topology design optimization of trusses [7], optimized design of two-dimensional structures [8], structural topology optimization [9], and structural damage detection [10]. For example, Rajeev and Krishnamoorthy [6] used GA for the optimization of the design of truss systems using discrete variables. The proposed GA implementation is tested with trusses of 3, 10, 25, and 160 elements. Rajan [7] uses GA to optimize the size shape and topology of trusses using continuous and discrete variables. Camp et al. [8] proposed to combine finite element analysis and GA for the design of discrete structural systems. The implementation of GA has not only been in the area of structural design. The method has also been used to update numerical models based on experimental data. Chou and Ghaboussi [10] used GAs to minimize the error between the natural frequencies of a numerical model and those found experimentally with the goal of identifying damage in the structure.

There are several well-established operators in genetic algorithms such as initialization, evaluation, selection, crossover, and mutation as shown in Figure 1 [11]. The initialization operator generates the individual of the first population based on a random number generator. The selection operator selects pairs of individuals (parents) based on their fitness values to breed a new generation. A variety of methods to select pairs of individuals such as roulette wheel selection, tournament selection, and rank selection have been proposed in the literature [11]. The crossover operator mates the pairs (parents) with a crossover probability to form a new offspring. There are also several methods for crossover of real parameter genes such as arithmetic crossover [12], BLX-a crossover (Blend crossover) [13], linear BGA (breeder genetic algorithm) crossover [14], and Wright's heuristic crossover [15]. The mutation operator helps to keep diversity in the population by randomly changing genes of some individuals. Mutation is usually an option of a GA described by the percentage of individuals that will be selected for mutation on each generation. An elite population operator is applied to keep the best individual from generation to generation [16].

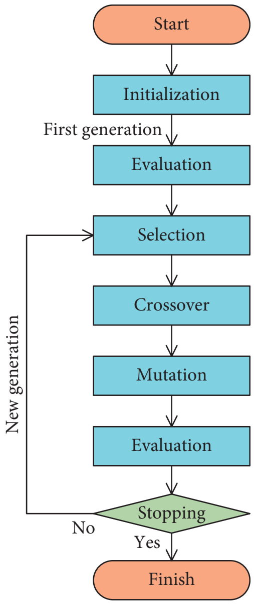


Figure 1: Traditional genetic algorithm flow chart.

This work is inspired by modeling to generate alternatives (MGA) [17] and hop, skip, and jump (HSJ). Brill et al. [17] presented the HSJ method as a technique to identify a small number of multiple solutions in optimization problems. The premise is that fitness functions in engineering applications might not include all the parameters involved in the problem. This is particularly true in complex problems spanning social, economic, and ethical dimensions. The analyst or engineer can make a decision on which solution to accept or reject based on his/her own experience and other information that might not be incorporated in the mathematical model. HSJ was originally used for complex or incompletely defined problems that have no exact solution. HSJ is applied to finite element model updating [18], sustainable multi-objective land-use allocation [19], and reducing epistemic uncertainty [20].

The identification of local optima using GA has been addressed in the literature (Maulik and Bandyopadhyay [21]; Streichert et al. [22]). Most of these techniques use a distance between the genes of individuals and the value of the fitness function as a way to determine if an individual is a local optimum or not. However, this does not take into consideration the shape of the fitness function, and local optima that are not "true" local optima can be identified incorrectly. Consider for example the maximization of the objective function $f(x)$ shown in Figure 2. A GA focusing in identifying local optima could identify S1, S2, and S3 as plausible local maxima. S1 is the global maxima and S2 is a "true" local maxima. However, S3 could also be identified because it is at a similar distance from the global maxima than S2, although it has a close fitness value to S2. The only way to determine that S3 is not a "true" local maximum is to study the shape of the objective function.

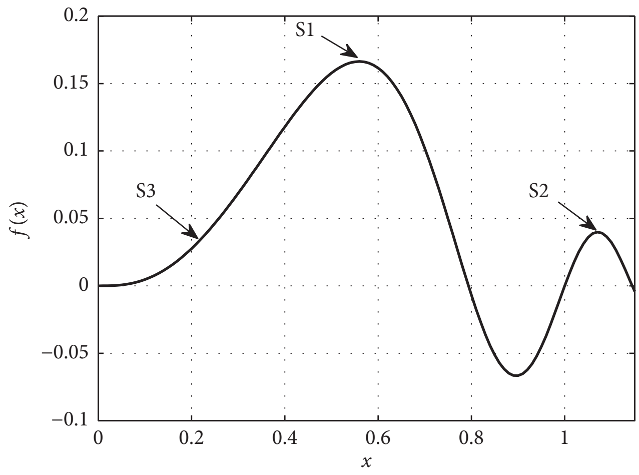


Figure 2: Plausible optima in a 1D objective function.

In this paper, a modified genetic algorithm called multi-solution genetic algorithm (MsGA) is proposed that identifies local optima by studying the shape of the fitness function. In addition, a section (cut) method is proposed here to identify "true" local optima which are used in the selection operator of the proposed GA implementation. As a result, MsGA can easily find multiple solutions which are ignored by the traditional genetic algorithms; however, those local solutions can be the expected solution alternatives from the design engineers' point of view.

2. Multi-Solution Genetic Algorithm (MsGA)

The multi-solution genetic algorithm (MsGA) proposes adding a new clustering operator to the traditional GA. The objective of this new operator is to find local minima in addition to the global solution. This operator includes a section method to distinguish between true and fictitious local optimal points. Figure 3 shows a flow diagram describing MsGA.

2.1. Clustering Operator. The goal of this operator is to identify solutions that have a similar performance of the objective function but are located in different parts of the solution space giving the analyst solutions that are physically different to choose. First, the individuals are sorted based on the fitness value. The best performer of the whole population is considered as the lead individual of cluster 1 (corresponding to the global optimum for the current generation). New clusters are assigned if the i-th individual complies with two criteria: the Euclidean distance between the location of the i-th individual and all previously found minima is greater than a user defined (Δ); and the value of the objective function is close to the best value of the objective function found so far. A user defined parameter α is used to perform the comparison. The minimum Euclidean distance (Δ) is also called minimum solution distance, and it allows the analyst to specify how far the multiple solutions prevent infinite solution around any possible solution. Each cluster can be visualized as a hypersphere centered at the local minima and with radius equal to the Euclidean distance (Δ). If two or more clusters overlap, that area belongs to the cluster that has local minima with the best performing individual (either higher fitness function for maximization or lower fitness function for minimization problems). The fitness function threshold is used to limit the fitness value of new local optima. Consider the minimization of one parameter function $f(\theta)$ shown in Figure 4 as an illustrative example. The value of the fitness function threshold is defined f_α is a constant defined by the user. In cases where the fitness function has a physical meaning, the analyst can specify a set value for f_α depending on the requirements of the optimization. The circles in Figure 4 represent individuals and the number inside each circle indicates their rank after sorting them using the fitness function. The fourth individual (Λ_4) is not considered the lead of a new group because the Euclidean distance to Λ_1 is less than Δ. However, Λ_2 becomes the lead of a new cluster because $f(\theta_2) \le f_\alpha$ and $\|\theta_2 - \theta_1\| \ge \Delta$, where $\| : \|$indicates Euclidean distance. Individual is in a local minimum, but it is not accepted because $f(\theta_6) > f_\alpha$.

2.2. Section Method. The algorithm can incorrectly identify points that do not correspond to a local optimum in the objective function if the values of α and Δ are not set correctly. For example, consider that the Euclidean distance (Δ) in the previous example is set to half of what is shown in Figure 4. The fourth individual would become a lead of a new local optima because $\|\theta_1 - \theta_4\| > \Delta$ and $f(\theta_4) < f_\alpha$. However, these two individuals represent the same minima because the slope between them does not change. An iterative process can be performed to eliminate this problem. This is done by "cutting" or creating a section of the objective function between two possible solutions and searching for changes of slope along the cut. In other words, a change in the slope in a section of the objective function passing through these individuals should be identified to consider these two points as two local minima. The cut is defined by the following equation:

$$\overline{u}_{\mathrm{S1-S2}} = \frac{\theta_{\mathrm{S2}} - \theta_{\mathrm{S1}}}{\left\|\theta_{\mathrm{S2}} - \theta_{\mathrm{S1}}\right\|}, \qquad (1)$$

where $\overline{u}_{\mathrm{S1-S2}}$ is the unit vector between solution S1 and S2, X_{Si} is the vector of parameters of the i-th possible solution, and $\| : \|$ indicates norm. Points between the two possible solutions can be obtained using the following equation:

$$X_n = X_{\mathrm{S1}} + n \times \overline{u}_{\mathrm{S1-S2}} \times \left\|\theta_{\mathrm{S2}} - \theta_{\mathrm{S1}}\right\|, \qquad (2)$$

where n is a number between zero and one. Higher numbers of cut points not also increase the accuracy but also the computational time. The cut method is not used for every single generation to reduce the computational time. The method is applied at p intervals as defined by the user.

3. Numerical Validation

Two problems defined by two variable functions are used in this section to validate the proposed algorithm. Two variable functions are used to be able to plot the values of the objective function and verify that the method is working

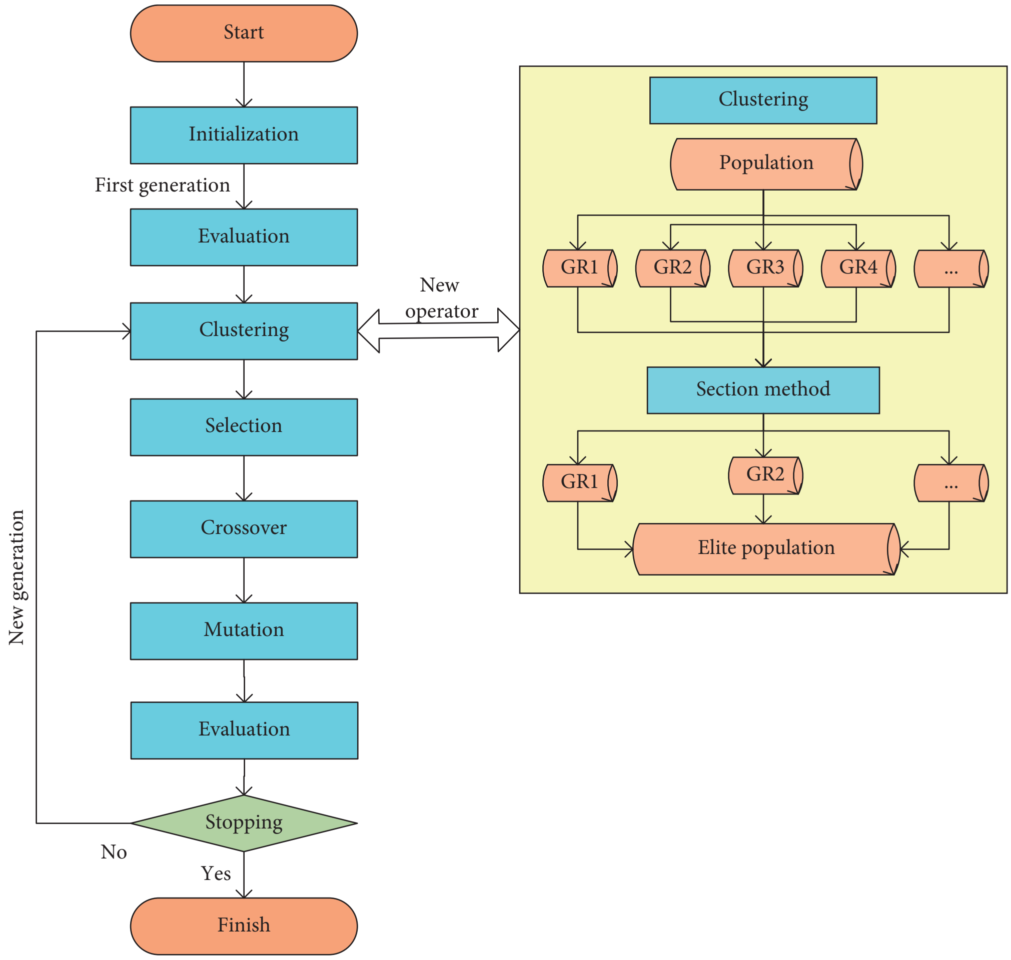


FIGURE 3: Multi-solution genetic algorithm flow chart.

correctly. The first problem is used to look at the effects of the algorithm parameters and the section method. The second problem is used to explore the behavior of the algorithm when the objective function has a large number of plausible solutions. The function for the first validation problem is obtained by combining several Gaussian-like functions defined by the following equation:

$$f(\theta_1, \theta_2) = \sum_{i=1}^{n} h_i \times e^{\left[(1/2)\left(\theta_1 - \mu_{1,i}/\sigma_{1,i}\right)^2 - (1/2)\left(\theta_2 - \mu_{2,i}/\sigma_{2,i}\right)^2\right]}, \quad (3)$$

where μ is a 2 by n matrix defining the location of the n maxima and σ is a 2 by n matrix that defines the shape of each maxima. In this equation, h represents a vector of size n which is used to modify the value of the objective function for each maximum point. In this example, four peaks are defined by the following matrices:

$$\mu = \begin{bmatrix} 1.5 & 0 & -1.5 & -2 \\ 0 & -1 & 1.5 & -2 \end{bmatrix},$$
$$\sigma = \begin{bmatrix} 0.4 & 0.4 & 1 & 0.4 \\ 1.5 & 0.4 & 1 & 0.4 \end{bmatrix}, \quad (4)$$
$$h = \begin{bmatrix} 2 & 1.6 & 1.8 & 0.6 \end{bmatrix}.$$

A surface plot of the objective function is shown in Figure 5. Notice that the local maxima are not exactly located at μ because of the interaction between different terms when summation is made. However, they are close enough for validation purposes. Even though MGA can handle nonsmooth functions with many variables, this example helps us investigate the capabilities of the technique. Selection of these two parameters is critical to the performance of the algorithm and usually requires some experience from the analyst. For example, a value of Δ can result in the identification of local maxima in the solution space that might not be of interest in engineering applications because they are physically similar and have a small difference in the fitness value. To illustrate this point, consider that the maximum fitness threshold is chosen as 70% and the minimum Euclidean distance as 5%. In addition, consider that the cut method is not used. A total of 12 maxima are found as shown in Figure 6. Solution 1 is the global maximum; solutions 4 and 8 are the local maxima. The rest of points shown in Figure 6 refer to these maxima (e.g., points 2, 3, 7, and 10 belong to solution 1 (global maximum); points 5, 6, 9, 11 and 12 to point 4 (local maximum)).

Figure 7 shows the cut between solutions 1 and 4 and between solutions 1 and 7. The proposed technique was able

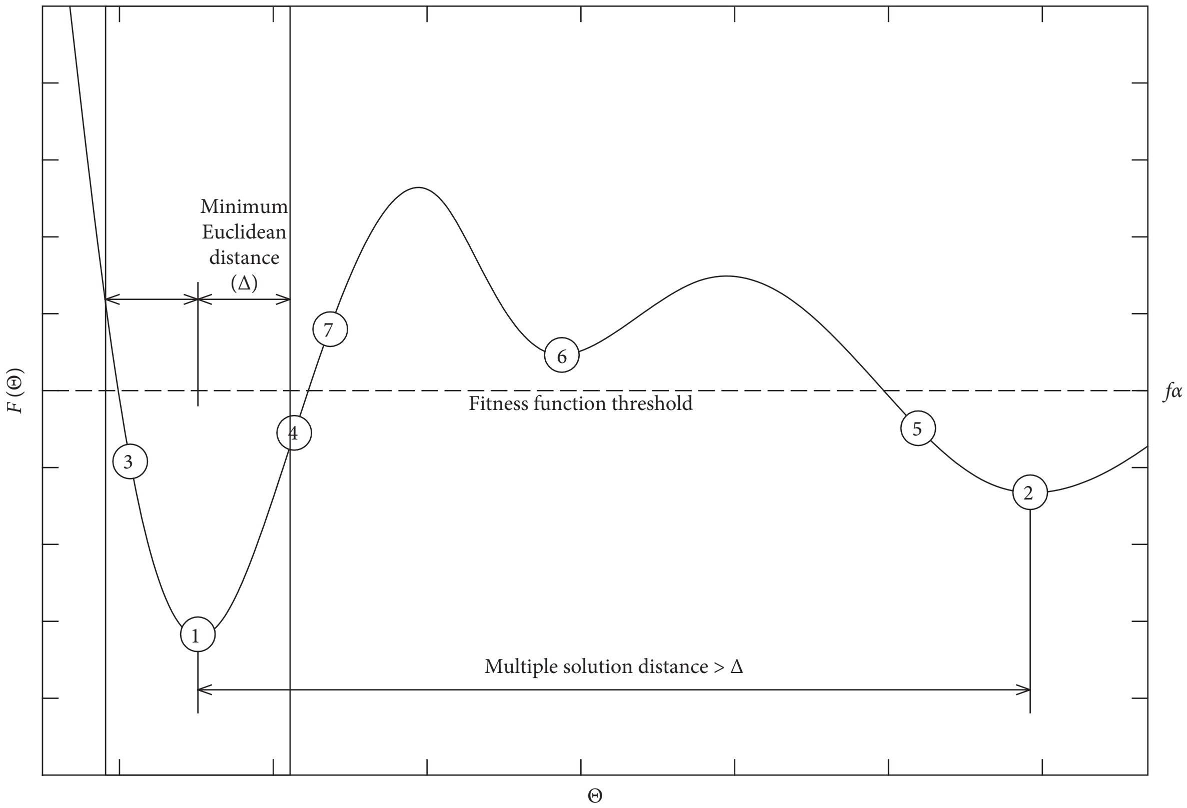


Figure 4: Clustering criteria.

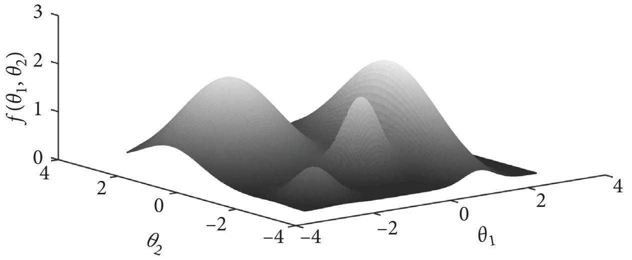


Figure 5: Surface plot for 2 parameters.

to find all maxima (Figure 8) when the minimum Euclidean distance (Δ) was set to 5% and the cut method is used. The numerical values for each maximum are shown in Table 1. The local maximum located at point (−2, 2) is not detected by the algorithm because the change of the objective function between the global maximum and this local maximum is bigger than the one which is specified at the beginning ($\alpha = 70\%$). In other words, it can be mentioned that this maximum is not of interest for the user.

Function for the second numerical validation is defined by a trigonometric equation previously used by Caicedo and Yun [23].

$$f(\theta_1, \theta_2) = \sin(19 \times \pi \times \theta_1) + \frac{\theta_1}{1.7} + \sin(19 \times \pi \times \theta_2) + \frac{\theta_2}{1.7}. \tag{5}$$

The function has a total of 100 maxima in the range of $\theta_1 = [0, 1]$ and $\theta_2 = [0, 1]$. Although it is unlikely to obtain an engineering problem with this number of local optima, this problem has been used to test similar other algorithms that identify local maxima. MsGA was run for 100 generations with 5000 individuals. Five percent of the population was considered elite between generations and it was not subjected to mutation. Fifteen percent of the population was mutated. These parameters are chosen because a large number of local maxima are expected in the fitness function. Particularly, the population size is higher than the results obtained from the first test. If the population size is not large enough, then some local solutions could be missed. The program was able to identify all local maxima as shown in Figure 9.

4. Implementation

MsGA is used to solve the classic 10-bar truss problem, as shown in Figure 10. This problem has been solved by other techniques such as genetic search methods [24] and a

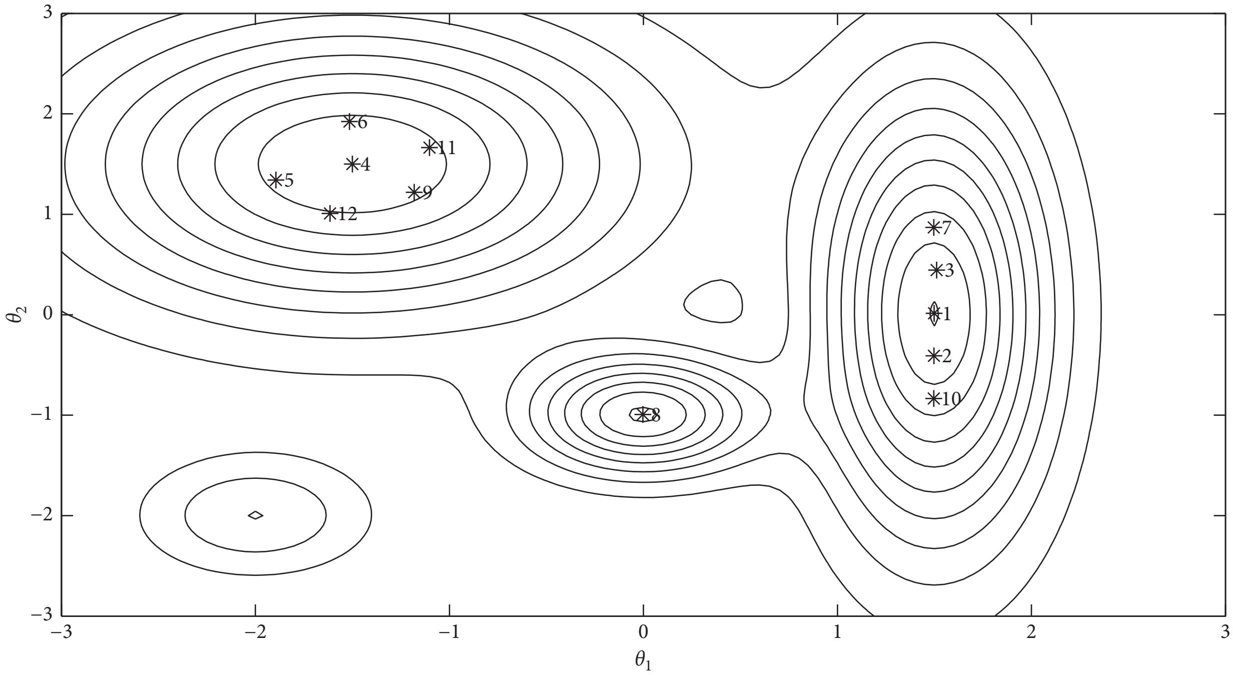


Figure 6: Results using $\alpha = 70\%$ and $\Delta = 5\%$.

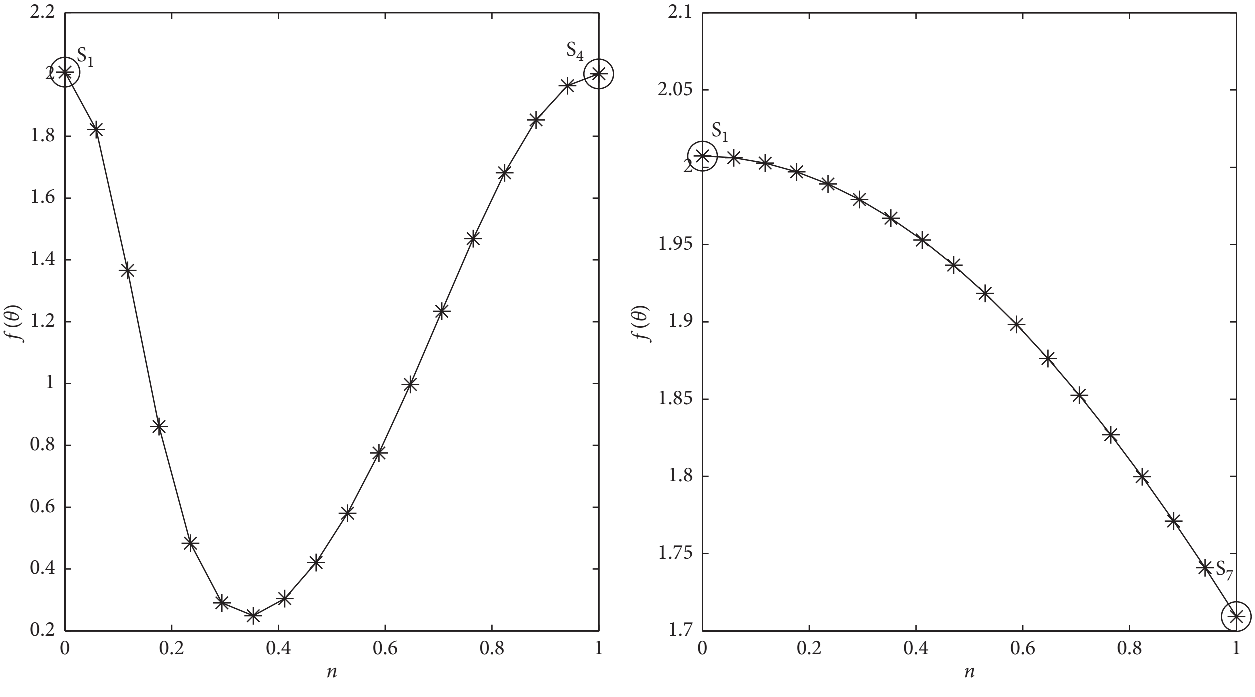


Figure 7: Cut between optima 1 and 4 (left) and optima 1 and 7 (right).

heuristic particle swarm method [25]. The problem consists in finding the cross-sectional area of the structural members that produce the minimum weight of the structure subject to a maximum stress on any truss element and minimum displacement at the node P. A force load $F = 100$ kips was applied at two different nodes as shown in Figure 10. The objective function is defined by

$$\min f(x) = \sum_{i=1}^{10} \rho_i l_i x_i, \tag{6}$$

where ρ_i is the density of element i, l_i is the length of element i, and x_i is the area of element i. The density of the elements is considered as 0.1 lb/in^3. The objective function is subject to

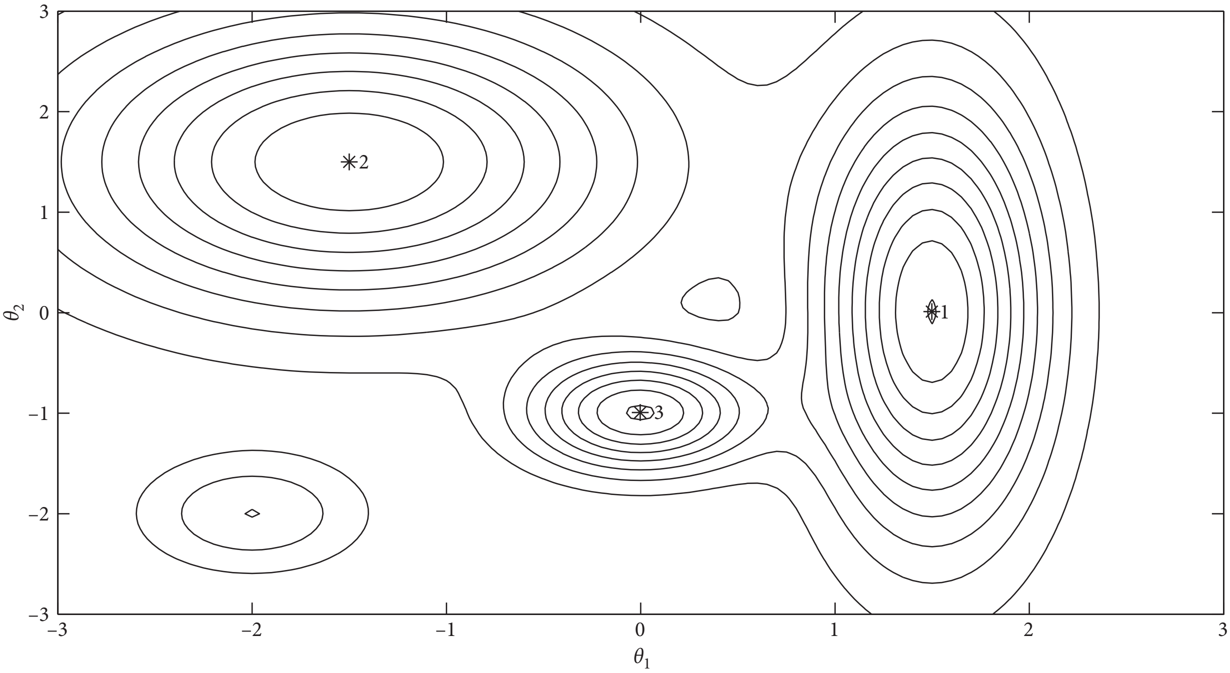


Figure 8: Multi-solutions after the cut method.

Table 1: Multi-solutions after the cut method.

No	$f(\theta_1, \theta_2)$	θ_1	θ_2
1	2.0066	1.4984	0.0107
2	1.8000	−1.5000	1.5000
3	1.6273	−0.0026	−0.9934

$$g_i(x) = \frac{\sigma_i}{\sigma_{\text{fea}}} - 1 \leq 0, \quad i = 1 \ldots 10,$$
$$g_{11} = \frac{\delta_P}{\delta_{P\text{fea}}} - 1 \leq 0, \tag{7}$$

where σ_i is the normal stress in element i and σ_{fea} is the maximum allowable stress defined as 25,000 psi in this paper. The variable δ_P is the vertical deflection of node P, and $\delta_{P\text{fea}}$ is the associated maximum allowable deflection defined here as 2 in. Young's modulus of the structure is $E = 10{,}000$ ksi, and dimension a in Figure 10 is 360 in. The cross-sectional area of each element is considered between 0.1 in^2 and 40 in^2. This is $0.1\,\text{in}^2 \leq x_i \leq 40\,\text{in}^2$, $i = 1 \ldots 10$.

MsGA was run for maximum 500 generations with 50,000 individuals. Five percent of the population was considered elite between generations, and it was not subjected to mutation. Fifteen percent of the population was mutated. These parameters are chosen because of the number of variables and limits of variables. The minimum Euclidean distance Δ was set to 5% and fitness function threshold was set to 8500. The MsGA results are shown in Table 2.

In this example, a total of 3 solutions were obtained which can be considered physically plausible. Although global solution has less weight than local solutions, second solution could be more efficient for construction. Building up 7th element may be easier than building up 9th element, because 7th element is closer to the support point. While increasing the cross section area of 7th element, the cross section area of 9th element is decreasing in second solution. Therefore, an expert may choose second solution due to labor efficiency and ease of manufacturing.

5. Discussion and Comparison

The difference between TGA and MsGA is that while TGA searches global optimum, MsGA searches global and local optima. When the aim of algorithm is just to find the global optimum, MsGA has some advantages and disadvantages against TGA. Since MsGA goes through more computational steps to find local optima by using clustering operator, the computational time increases as a result. Even though MsGA has some disadvantages for finding global optima, it has also some advantages. For instance, most of stochastic algorithms and evolutionary algorithms can easily get stuck in a local optimum while looking for a global optimum. Since MsGA searches both global and local solutions together, there is no problem for this method to get stuck or trapped in local optimum solution.

In addition, it should be clarified here that since stochastic methods work in a random nature, it would not be fair to compare such algorithms with MsGA or each other. Because for a stochastic algorithm, results always differ from one program run to another. Naturally, there are too many parameters that directly affect the results, such as initial population size, generation size, computational time, search space, domain, etc. However, among the researchers, the most popular evolutionary algorithm is thought to be TGA. In this paper, a new approach to TGA is proposed with additional operators so that MsGA can be able to find local

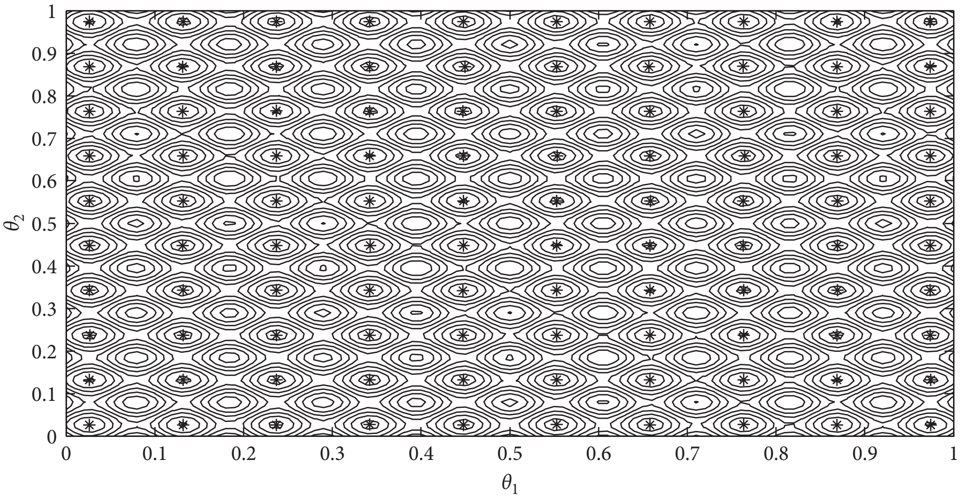


Figure 9: Multiple solutions for two variable functions with 5000 population size.

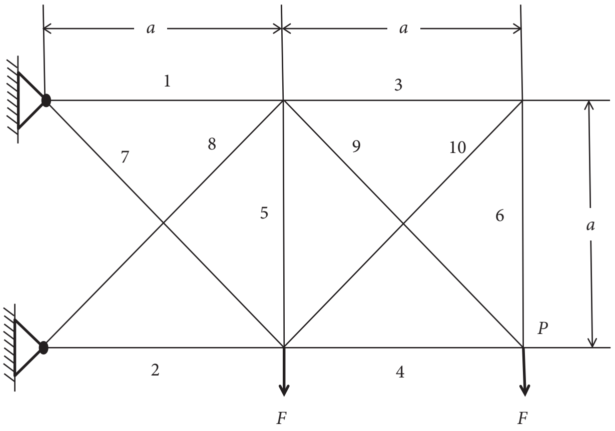


Figure 10: Ten-bar truss.

Table 2: Results with MGA.

x_i (in^2)	Solution 1	Solution 2	Solution 3
X_1	30.71	29.53	30.09
X_2	21.69	23.74	24.87
X_3	0.27	2.15	0.89
X_4	15.91	16.57	21.88
X_5	0.21	0.21	8.09
X_6	0.55	2.52	0.91
X_7	7.33	12.96	9.12
X_8	21.72	24.38	24.27
X_9	21.22	16.11	19.89
X_{10}	0.42	3.25	1.40
min f(lb)	5078.5	5578.4	5908.1

and global optimum solutions for alternative results for engineering problems.

6. Result and Conclusion

This paper presents a modified genetic algorithm (MsGA) which differs from traditional genetic algorithm (TGA). MsGA makes it possible to find real multiple solutions after each generation unlike TGA. Generally, optimization problems are represented by a mathematical model which is also used as an objective function during the optimization process. The proposed MsGA has been applied on a known engineering benchmark problem. For the truss optimization problem, different optimal solutions are captured by the algorithm which can be further analyzed by an expert to be decided for site implementation regarding the labor efficiency, ease of manufacturing cost issues etc. It is concluded that MsGA is successful at finding different optimal solutions to a unique problem.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Subramanian, A. S. S. Sekar, and K. Submaranian, "African wild dog algorithm," *International Journal of Soft Computing*, vol. 8, pp. 163–170, 2013.

[2] B. Doğan and C. Olmez, "Vortex search algorithm for the analog active filter component selection problem," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 9, pp. 1243–1253, 2015.

[3] A. Kaveh, R. Sheikholeslami, S. Talatahari, and M. Keshvari-Ilkhichi, "Chaotic swarming of particles: a new method for size optimization of truss structures," *Advances in Engineering Software*, vol. 67, pp. 136–147, 2014.

[4] G. Dhiman and V. Kumar, "Emperor penguin optimizer: a bio-inspired algorithm for engineering problems," *Knowledge-Based Systems*, vol. 159, pp. 20–50, 2018.

[5] M. Tabassum and K. Mathew, "A genetic algorithm Analysis towards optimization solutions," *International Journal of Digital Information and Wireless Communications*, vol. 4, no. 1, pp. 124–142, 2014.

[6] S. Rajeev and C. S. Krishnamoorthy, "Discrete optimization of structures using genetic algorithms," *Journal of Structural Engineering*, vol. 118, no. 5, pp. 1233–1250, 1992.

[7] S. D. Rajan, "Sizing, shape, and topology design optimization of trusses using genetic algorithm," *Journal of Structural Engineering*, vol. 121, no. 10, pp. 1480–1487, 1995.

[8] C. Camp, S. Pezeshk, and G. Cao, "Optimized design of two-dimensional structures using a genetic algorithm," *Journal of Structural Engineering*, vol. 124, no. 5, pp. 551–559, 1998.

[9] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya, and K. Saitou, "Continuum structural topology design with genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 339–356, 2000.

[10] J.-H. Chou and J. Ghaboussi, "Genetic algorithm in structural damage detection," *Computers and Structures*, vol. 79, no. 14, pp. 1335–1353, 2001.

[11] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.

[12] Z. Michalewicz, *Genetic algorithms+data structures= evolution programs*, Springer-Verlag, Berlin and New York, 3rd edition, 1996.

[13] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, 1993.

[14] D. Schlierkamp-Voosen, "Strategy adaptation by competition," in *Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing*, pp. 1270–1274, Aachen, Germany, September 1994.

[15] A. H. Wright, "Genetic algorithms for real parameter optimization," *Foundations of Genetic Algorithms*, pp. 205–218, 1991.

[16] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, no. 6, pp. 731–747, 1996.

[17] E. D. Brill, S.-Y. Chang, and L. D. Hopkins, "Modeling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning," *Management Science*, vol. 28, no. 3, pp. 221–235, 1982.

[18] B. A. Zárate and J. M. Caicedo, "Finite element model updating: multiple alternatives," *Engineering Structures*, vol. 30, no. 12, pp. 3724–3730, 2008.

[19] A. Ligmann-Zielinska, L. Church Richard, and P. Jankowski, "Spatial optimization as a generative technique for sustainable multiobjective land-use allocation," *International Journal of Geographical Information Science*, vol. 22, no. 6, pp. 601–622, 2008.

[20] J. M. Caicedo and B. A. Zárate, "Reducing epistemic uncertainty using a model updating cognitive system," *Advances in Structural Engineering*, vol. 14, no. 1, pp. 55–65, 2016.

[21] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.

[22] F. Streichert, G. Stein, H. Ulmer, and A. Zell, "A clustering based niching EA for multimodal search spaces," *Lecture Notes in Computer Science*, vol. 2936, pp. 293–304, 2004.

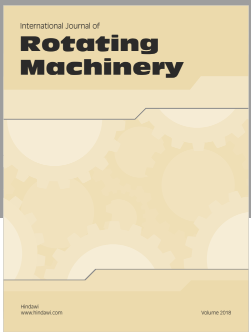
[23] J. M. Caicedo and G. J. Yun, "A novel evolutionary algorithm for identifying multiple alternative solutions in model updating," *Structural Health Monitoring: An International Journal*, vol. 10, no. 5, pp. 491–501, 2010.

[24] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Structural optimization*, vol. 4, no. 2, pp. 99–107, 1992.

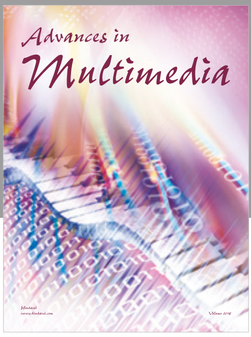
[25] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers and Structures*, vol. 87, no. 56, pp. 267–283, 2009.

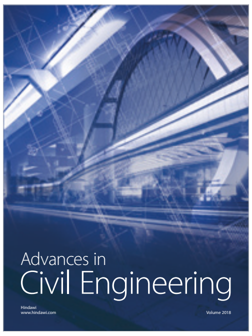
Journal of
Engineering

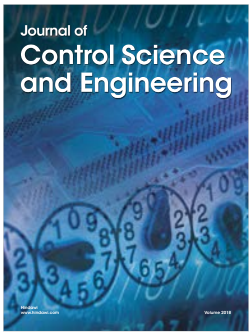
The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

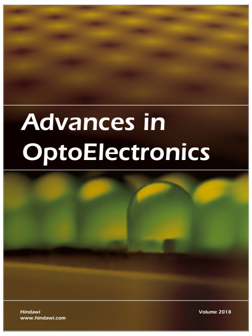

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

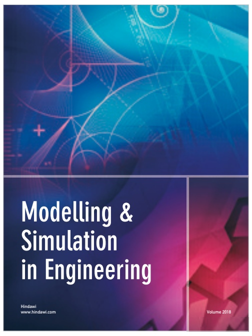
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

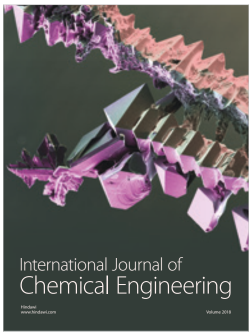

Advances in
OptoElectronics

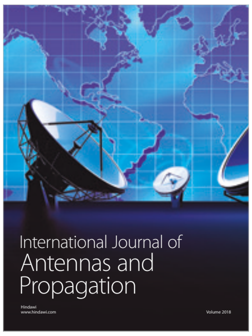
VLSI Design

International Journal of
Navigation and
Observation

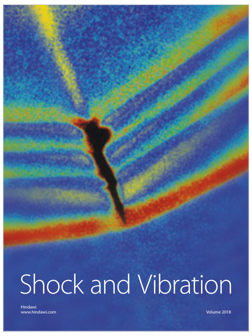

Modelling &
Simulation
in Engineering

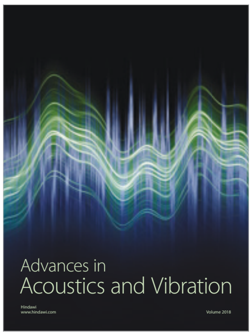
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration