INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2012; 2(4):479-487

AN EFFICIENT METAHEURISTIC ALGORITHM FOR ENGINEERING OPTIMIZATION: SOPT

O. Hasançebi[*, †] and S. Kazemzadeh Azad
Middle East Technical University, Department of Civil Engineering, Ankara, Turkey

ABSTRACT

Metaheuristic algorithms are well-known optimization tools which have been employed for solving a wide range of optimization problems so far. In the present study, a simple optimization (SOPT) algorithm with two main steps; namely exploration and exploitation, is provided for practical applications. Aside from a reasonable rate of convergence attained, the ease in its implementation and dependency on few parameters only are among the advantageous characteristics of the proposed SOPT algorithm. The efficiency of the developed algorithm is investigated through engineering design optimization problems and the results are reported. The comparison of the numerical results with those of other metaheuristic techniques demonstrates the promising performance of the algorithm as a robust optimization tool for practical purposes.

Received: 5 July 2012; Accepted: 30 September 2012

KEY WORDS: metaheuristics; optimization algorithm; practical optimization; engineering design

1. INTRODUCTION

Metaheuristic search techniques, such as simulated annealing (SA) [1], genetic algorithm (GA) [2], evolution strategies (ESs) [3], particle swarm optimization (PSO) [4], ant colony optimization (ACO) [5, 6], etc., which are generally developed based on natural phenomena [7], have become the popular optimization techniques of the recent years due to their capability of finding promising solutions for complicated optimization problems as well as their independency to the derivatives of objective functions. Further, metaheuristics can

[*]Corresponding author: O. Hasançebi, Department of Civil Engineering, Middle East Technical University, 06800, Ankara, Turkey
[†]E-mail address: oguzhan@metu.edu.tr

handle both discrete and continuous variables and can be applied to a wide range of optimization problems, effectively. Basically, both trajectory and population based metaheuristic approaches aim to locate the global optimum in the solution space through random moves. The key difference between the algorithms is in the way that they propose the next move in the solution space. Here, the utilized strategies and mechanisms for proposing more reliable moves become crucial. This motivates developers of optimization algorithms to find more efficient methodologies for originating robust optimization algorithms. However, sometimes this results in complicated approaches which are difficult to understand and implement. Hence, this study is an attempt to provide a simple and efficient methodology for engineering optimization purposes.

Generally, in population based algorithms a population of candidate solutions is employed to seek the optimum solution based on specific strategies utilized in each technique. Since the working principals of these techniques are somewhat identical, genetic algorithms (GAs) are outlined here as an instance. In GAs, the individuals of a population undergo an evolutionary process which results in surviving of the fittest individuals of the population during the optimization. The crossover and mutation operators as well as selection and reproduction mechanisms are among the well-known tools commonly used in different variants of the GAs for improving the quality of solutions during the iterations. The last iteration of the GAs is expected to include optimum or reasonably near-optimum individuals that survive due to their competitive quality or fitness. Due to their efficiency as well as the ease of understanding and implementation, GAs have been widely employed for solving various optimization problems so far.

In the recent years, novel optimization techniques, such as harmony search (HS) method [8], big bang-big crunch (BB-BC) [9], artificial bee colony (ABC) algorithm [10], charged system search (CSS) [11], firefly algorithm (FA) [12], etc. have been developed based on different search strategies to lessen the burden of locating the global optimum in complicated optimization problems. In the present study, a simple optimization (SOPT) algorithm is proposed for handling engineering optimization problems. The proposed technique provides an efficient strategy for investigation of the solution space to locate the optimum or a competitive near-optimum solution. The SOPT algorithm and related formulations are described in the next section. The third section of the paper covers performance evaluation of the proposed approach through benchmark instances. The last section provides a clear conclusion of the study.

2. DESCRIPTION OF THE ALGORITHM

In the present study, the main concern is to provide a simple and efficient formulation which can be employed in engineering optimization applications. The SOPT algorithm is a population based algorithm composed of two main steps; namely exploration and exploitation steps. In this algorithm the exploration and exploitation steps are performed one after another based on the following strategy. Here, the exploration step is carried using Equation (1), where the *i*-th parameter of a new candidate solution $x_{(new)}$ is generated as follows:

$$x_{new(i)} = x_{best(i)} + \lambda_1 \times R_{(i)} \quad (1)$$

In Equation (1), λ_1 is a positive constant, and $R_{(i)}$ is a normally distributed random number with a mean zero and a standard deviation $\sigma_{R(i)}$. For each *i*-th parameter, a standard deviation $\sigma_{R(i)}$ is computed at each iteration by calculating the standard deviation of the respective parameter in all the members of the population. This strategy of choosing the standard deviation for the normal distribution, which is initially employed in [14], is schematically shown in Figure 1 for a small population of four candidate solutions where each candidate solution is composed of *NT* variables.

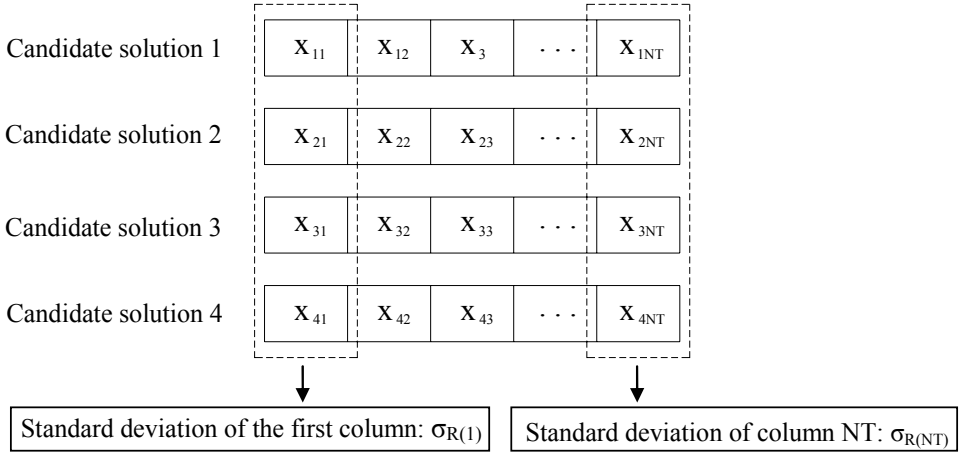

Figure 1. Schematic representation of $\sigma_{R(i)}$ for four solutions in each step of SOPT

A similar equation to the aforementioned one is also employed for exploitation step of the SOPT as follows:

$$x_{new(i)} = x_{best(i)} + \lambda_2 \times R_{(i)} \quad (2)$$

where λ_2 is equal to $0.5\lambda_1$ by which, in comparison to the exploration stage, the generation of new candidate solutions is more probable in the vicinity of the best solution. This results in a more exploitative investigation strategy in this step. It is worth mentioning that the described exploration and exploitation formulations are considered based on a relative comparison between the aforementioned two equations in terms of the generated candidate solutions and do not imply a general definition for these two concepts.

In each step of SOPT, after the new candidate solutions are generated, the worst members of the population are replaced with the better new ones. A similar evolutionary scheme can be found in ESs [3]. In SOPT the aforementioned process continues repeatedly until a predefined termination condition is met. For the sake of clarity the SOPT algorithm can be outlined as follows:

Step 1: A population of randomly proposed candidate solutions is generated (random initial exploration).

Step 2: The population members are evaluated based an objective function.

Step 3: The best candidate solution among the population is determined.

Step 4: The standard deviation of each column of population is calculated to use in the next step.

Step 5: The exploitation step initiates by generating the new candidate solutions using Eq. (2).

Step 6: Newly generated candidate solutions are evaluated.

Step 7: The worst members of the population are replaced with the better new ones.

Step 8: The best candidate solution among the population is determined.

Step 9: The standard deviation of each column of population is calculated to use in the next step

Step 10: The exploration step initiates by generating the new candidate solutions using Eq. (1).

Step 11: Newly generated candidate solutions are evaluated.

Step 12: The optimization is repeated from step 3 until a termination criterion, such as maximum iteration number, is satisfied.

Apart from the efficiency of the SOPT algorithm, which is illustrated in the next section through numerical examples, the ease in its implementation and dependency on few parameters only are some advantageous characteristics of the SOPT algorithm.

3. NUMERICAL EXAMPLES

This section includes two well-known benchmark engineering optimization instances employed for performance evaluation of the proposed SOPT algorithm. For both of the examples,50 candidate solutions are used to represent the population, the value of parameter λ_1 is taken as 2, and the termination condition is set to the 10,000 objective function evaluations. The optimum solutions attained using the SOPT algorithm are compared to the previously reported results in the literature.

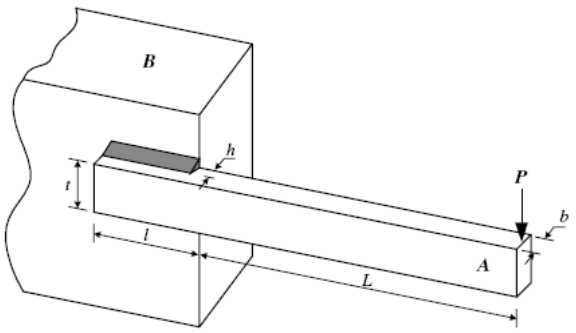

Figure 2.Welded beam structure [15].

3.1 Example 1: welded beam design

In the first example the design optimization of the welded beam (A), shown in Figure 2, is carried out. This benchmark problem is considered in numerous studies so far [15-22]. In this problem, the objective is to find the best set of design variables to minimize the total fabrication cost of the structure subject to shear stress (τ), bending stress (σ), buckling load (P_c), and end deflection (δ) constraints. Assuming $x_1 = h$, $x_2 = l$, $x_3 = t$, and $x_4 = b$ as the design variables, the mathematical formulation of the problem can be posed as follows [22]:

Find

$$x = \{x_1, x_2, x_3, x_4\} \tag{3}$$

to minimize

$$Cost(\mathbf{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2) \tag{4}$$

subject to

$$\begin{aligned}
&g_1(x) = \tau(x) - \tau_{\max} \le 0 \\
&g_2(x) = \sigma(x) - \sigma_{\max} \le 0 \\
&g_3(x) = x_1 - x_4 \le 0 \\
&g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \le 0 \\
&g_5(x) = 0.125 - x_1 \le 0 \\
&g_6(x) = \delta(x) - \delta_{\max} \le 0 \\
&g_7(x) = P - P_c(x) \le 0
\end{aligned} \tag{5}$$

The bounds on the design variables are:

$$0.1 \le x_1 \le 2\,,\quad 0.1 \le x_2 \le 10\,,\quad 0.1 \le x_3 \le 10\,,\quad 0.1 \le x_4 \le 2 \tag{6}$$

where

$$\begin{aligned}
&\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\
&\tau' = \frac{P}{\sqrt{2}x_1 x_2} \qquad \tau'' = \frac{MR}{J} \qquad M = P\left(L + \frac{x_2}{2}\right) \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\
&J = 2\left\{\sqrt{2}x_1 x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \quad \sigma(x) = \frac{6PL}{x_4 x_3^2} \qquad \delta(x) = \frac{4PL^3}{E x_3^3 x_4} \\
&P_c(x) = \frac{4.013\sqrt{E(x_3^2 x_4^6/36)}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)
\end{aligned} \tag{7}$$

The constants in Eqs. (5) and (7) are chosen as follows: P = 6000 lb, L =14 in., E = 30×10^6 psi, $G = 12\times10^6$ psi, τ_{max} = 13600 psi, σ_{max} = 30000 psi, and δ_{max} = 0.25 in.

The optimum design of the welded beam is carried out using SOPT algorithm, and the best solution is found as x^*= {x_1, x_2, x_3, x_4} = {0.205729004650527,3.47050320445079, 9.03663339710796, 0.205729647805016} which yields an objective function value of Cost(x) = 1.72485498816014 for this example. Table 1 provides a comparison of this solution with the results of other optimization algorithms available in the literature. It is apparent form the table that SOPT algorithm finds a competitive solution using only 10,000 objective function evaluations which is considerably lesser than those of other approaches. Further, a statistical evaluation of 100 independent runs of the SOPT algorithm is tabulated in Table 2 considering the best, worst, average, and standard deviation (S.D.) of the obtained solutions.

Table 1. Comparison of the results for the welded beam design problem

Design variables	Lee and Geem [15]	Gandomi et al. [20]	Kazemzadeh Azad et al. [21]	SOPT
x_1	0.2442	0.2015	0.2054	0.20573
x_2	6.2231	3.562	3.4783	3.47050
x_3	8.2915	9.0414	9.0386	9.03663
x_4	0.2443	0.2057	0.2057	0.20573
$Cost(x)$	2.38	1.73121	1.72576	1.72485
No. of evaluations	110,000	50,000	20,000	10,000

Table 2. General performance of the SOPT algorithm in the welded beam design problem

Performance	Kazemzadeh Azad et al. [21]	SOPT
Best	1.72576	1.72485
Average	1.773	1.72491
Worst	2.1376	1.72570
S.D.	0.0824	0.0001

3.2. Example 2: design of a pressure vessel

Design optimization of the cylindrical pressure vessel capped at both ends by hemispherical heads (Figure 3) is considered as the second example [23]. The objective of optimization is to minimize the total manufacturing cost of the vessel based on the combination of welding, material and forming costs. The vessel will be designed for a working pressure of 3000 psi and a minimum volume of 750 ft^3regarding the provisions of *ASME* boiler and pressure vessel code. Here, the shell and head thicknesses should be multiples of 0.0625 in. The thickness of the shell and head is restricted to 2 in. The shell and head thicknesses are not to be less than 1.1 in. and 0.6 in., respectively. The solution variables of the problem are x_1 as the shell thickness (T_s), $x_2$as the spherical head thickness (T_h), $x_3$as the radius of cylindrical shell (R), and x_4 as the shell length (L). The problem formulation is as follows:

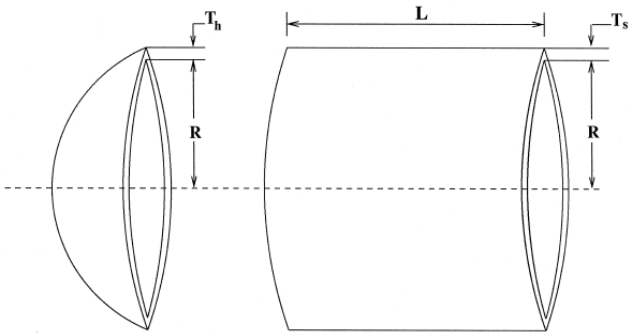

Figure 3. Spherical head and cylindrical shell of the pressure vessel [19]

Find

$$x = \{x_1, x_2, x_3, x_4\} \tag{8}$$

to minimize

$$Cost(x) = 0.6224x_3x_1x_4 + 1.7781x_3^2x_2 + 3.1611x_1^2x_4 + 19.8621x_3x_1^2 \tag{9}$$

subject to

$$\begin{aligned} g_1(x) &= 0.0193x_3 - x_1 \le 0 \\ g_2(x) &= 0.00954x_3 - x_2 \le 0 \\ g_3(x) &= 750 \times 1728 - \pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 \le 0 \\ g_4(x) &= x_4 - 240 \le 0 \end{aligned} \tag{10}$$

where the bounds on the discrete variables are as follows:

$$1.125 \le x_1 \le 2\,, \quad 0.625 \le x_2 \le 2 \tag{11}$$

Further, the bounds on the continuous variables, x_3 and x_4, are taken as:

$$10 \le x_3 \le 240 \quad , \; 10 \le x_4 \le 240 \tag{12}$$

For this benchmark example, the best solution attained from 100 independent runs of the SOPT algorithm is provided in Table 3. The SOPT algorithm locates a solution vector of $x^* = \{x_1, x_2, x_3, x_4\} = \{1.125, 0.625, 58.2901551776991, 43.6926585170488\}$ through only 10,000 objective function evaluations which results in an objective function value of $Cost(x)$ = 7199.35937506206 for this problem. Further, the general performance of the SOPT algorithm in 100 independent runs is given in Table 4.It is apparent from the results that SOPT algorithm is able to provide promising solutions with less objective function evaluations. This desirable characteristic of the SOPT algorithm would be more significant in case of engineering optimization problems which entail higher computational effort.

Table 3. Comparison of the results for the pressure vessel design problem

Design variables	Kazemzadeh Azad et al. [21]	SOPT
x_1	1.125	1.125
x_2	0.625	0.625
x_3	58.2895	58.2902
x_4	43.6964	43.6927
Cost(x)	7199.412	7199.359
No. evaluations	25,000	10,000

Table 4. General performance of the SOPT algorithm in the pressure vessel design problem

Performance	Kazemzadeh Azad et al. [21]	SOPT
Best	7199.412	7199.359
Average	7347.105	7208.215
Worst	9770.499	7342.977
S. D.	420.07	29.16

4. CONCLUSION

In the present study, the SOPT algorithm is proposed as a simple and efficient optimization technique for handling engineering optimization problems. The algorithm developed is composed of two main steps namely exploration and exploitation steps. The SOPT algorithm is a population based technique which follows a stochastic iterative procedure to locate the optimum or a reasonably near-optimum solution for a given optimization problem. Performance evaluation of the SOPT algorithm through benchmark design optimization examples reveals the efficiency of this technique in solving practical optimization problems. Although in the present study the algorithm is utilized only for solving engineering design optimization problems, SOPT algorithm can be easily employed for solving other types of optimization problems as well.

REFERENCES

1. Kirkpatrick S, Gerlatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;**220**:671–80.
2. Goldberg DE, Samtani MP. Engineering optimization via genetic algorithm. Proceeding of the Ninth Conference on Electronic Computation, ASCE 1986, pp. 471-82.
3. H.-P. Schwefel, "Numerical optimization of computer models", John Wiley & Sons, Chichester, UK, 1981.
4. Kennedy J, Eberhart R. Particle swarm optimization. In: IEEE International Conference on Neural Networks, IEEE Press, Vol. **4**, 1995;pp. 1942–1948.
5. Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colony. In:

Proceedings of the First European Conference on Artificial Life, USA 1991, pp. 134–142.

6. Dorigo M. Optimization, learning and natural algorithms, PhD thesis. Dipartimento Elettronica e Informazione, Politecnico di Milano, Italy 1992.
7. Yang X-S. *Nature-inspired Metaheuristic Algorithms*. Luniver Press; 2008.
8. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Comput Struct*, 2004; **82**:781–98.
9. Erol OK, Eksin I. New optimization method: Big Bang–Big Crunch. *Adv Eng Software*, 2006; **37**: 106–11.
10. D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
11. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. *Acta Mech*, 2010; **213**(3−4):267−289.
12. Yang XS. Firefly algorithms for multimodal optimization, in: *Stochastic Algorithms: Foundations and Applications* (Eds O. Watanabe and T. Zeugmann), SAGA 2009, *Lecture Notes in Computer Science*, 5792, Springer-Verlag, Berlin, 2009, pp. 169-178.
13. X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), *Studies in Computational Intelligence*, Springer Berlin, 284, Springer, 65-74 (2010).
14. Koohestani K, Kazemzadeh Azad S. An Adaptive Real-Coded Genetic Algorithm for Size and Shape Optimization of Truss Structures, in B.H.V. Topping, Y. Tsompanakis, (Editors), Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Civil-Comp Press, Stirlingshire, UK, Paper 13, 2009.doi:10.4203/ccp.92.13.
15. Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Meth Appl Mech Eng, 2005; **194**(36–38): 3902–33.
16. Siddall JN. Analytical design-making in engineering design. Prentice Hall, 1972.
17. Ragsdell KM, Phillips DT. Optimal design of a class of welded structures using geometric programming. *J Eng Ind Trans ASME*, 1976; **98**(3):1021–25.
18. Deb K. Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, 1991; **29**(11): 2013–15.
19. Coello Coello CA. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind*, 2000; **41**(2):113–27.
20. Gandomi AH, Yang XS, Alavi AH. Mixed variable structural optimization using firefly algorithm, *Comput Struct*, 2011; **89**(23–24): 2325–36.
21. S. Kazemzadeh Azad, O. Hasançebi, and O. K. Erol, Evaluating efficiency of Big-Bang Big-Crunch algorithm in benchmark engineering optimization problems, *Int J Optim Civil Eng*, 2011; **1**(3):495-05.
22. Rao SS. *Engineering Optimization: Theory and Practice*. John Wiley and Sons, 3rd edition, 1996.
23. Sandgren E. Nonlinear integer and discrete programming in mechanical design optimization, *Trans ASME J Mech Des*, 1990; **112**: 223–29.