Structural Engineering and Mechanics, *Vol. 60, No. 6 (2016) 1093-1117*
DOI: http://dx.doi.org/10.12989/sem.2016.60.6.1093

Optimal design of truss structures using a new optimization algorithm based on global sensitivity analysis

A. Kaveh[*1] and V.R. Mahdavi[2]

[1]*Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Tehran, Iran*
[2]*School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran*

(Received February 4, 2016, Revised October 8, 2016, Accepted October 18, 2016)

Abstract. Global sensitivity analysis (GSA) has been widely used to investigate the sensitivity of the model output with respect to its input parameters. In this paper a new single-solution search optimization algorithm is developed based on the GSA, and applied to the size optimization of truss structures. In this method the search space of the optimization is determined using the sensitivity indicator of variables. Unlike the common meta-heuristic algorithms, where all the variables are simultaneously changed in the optimization process, in this approach the sensitive variables of solution are iteratively changed more rapidly than the less sensitive ones in the search space. Comparisons of the present results with those of some previous population-based meta-heuristic algorithms demonstrate its capability, especially for decreasing the number of fitness functions evaluations, in solving the presented benchmark problems.

Keywords: global sensitivity analysis; meta-heuristic; single-solution; sensitivity indicator; truss structures

1. Introduction

The goal of engineering optimization problems is finding the best set of variables that fulfill all the design limitations having the lowest possible cost. Optimization algorithms are such techniques for solution of this type of problems. Hence, a vast number of algorithms have been developed to solve various problems in this field. These methods can generally be divided into two categories: deterministic and stochastic methods. The first ones are based on numerical linear and nonlinear programming methods that require the gradient information and usually applied to optimization problems for improving the solution around a starting point. On the other hand, the stochastic algorithms are suitable for global search due to their capability of exploring and finding promising regions in the search space by an affordable computational time (Gonzalez 2007, Talbi 2009, Yang 2010, Kaveh and Mahdavi 2015a),

These methods have some advantages and some drawbacks. The deterministic methods, for example, can obtain solution with higher convergence rate compared to the stochastic approaches since the former methods use the gradient information to obtain the minima. However, the

∗Corresponding author, Professor, E-mail: alikaveh@iust.ac.ir
[a]Ph.D. Student, E-mail: vahidreza_mahdavi@civileng.iust.ac.ir

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

estimation of gradient information can be either costly or even impossible for discrete design variables, and the optimum result obtained using these methods is completely dependent on the use of a good starting point. On the other hand, the meta-heuristic algorithms, as stochastic methods, are more general and can easily be implemented. Some examples of these methods are: Genetic algorithms (GA) of Holland (1975), Particle swarm optimization (PSO) of Eberhart and Kennedy (1995), Ant colony optimization (ACO) of Dorigo, Maniezzo *et al*. (1996), Big bang-big crunch (BB-BC) of Erol and Eksin (2006), Charged system search (CSS) of Kaveh and Talatahari (2010), Ray optimization (RO) of Kaveh and Khayatazad (2012), Dolphin echolocation (DE) of Kaveh and Farhoudi (2013), Min blast (MB) of Sadollah, Bahreininejad *et al*. (2012), Colliding Bodies Optimization (CBO) of Kaveh and Mahdavi (2014), Water evaporation optimization (WEO) of Kaveh and Bakhshpoori (2016), Whale Optimization Algorithm (WOA) of Mirjalili and Lewis (2016), Some applications of metaheuristic algorithms can be found in the work of Gholizadeh and Poorhoseini (2015) Gholizadeh, Gheyratmand *et al*. (2016), A meta-heuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploration (global search) and exploitation (local search) of the search space. Learning strategies are used to structure the information in order to find efficiently near-optimal solutions, Kaveh and Mahdavi (2015), one of the disadvantages of the meta-heuristic algorithms is that they do not consider the sensitivity information of design variables to the objective function to push the populations into new positions. In other word, in two subsequent optimization iterations, all design variables of a population have the same ranks for generating the new population. This makes the meta-heuristic algorithms slow to convergence into global optima. However, one can extract the sensitivity information of the current population, before generation process, to speedily guide the populations to a near-optimal solution.

In this paper, we introduce a new single-solution search optimizer, namely Global Sensitivity Analysis Based (GSAB) that uses a basic set of mathematical techniques, namely global sensitivity analysis (GSA), the sensitivity analysis (SA) studies the sensitivity of the model output with respect to its input parameters, Rahman (2011), This analysis is generally categorized as local SA and global SA techniques. While local SA studies sensitivity of the model output about variations around a specific point, the global SA considers variations of the inputs within their entire feasibility space (Pianosi and Wagener 2015, Zhai, Yang *et al*. 2014), One important feature of the GSA is Factor Priorization (FP), which aims at ranking the inputs in terms of their relative contribution to output variability. The GSAB comprises of a search optimization strategy and GSA-driven procedure, where the search space is guided by ranking the design variables using the GSA approach, resulting in an efficient and rapid search. The proposed algorithm can be studied within the family of search algorithms such as the Random Search (RS) of Rastrigin (1963), Pattern Search (PS) of Hooke and Jeeves (1961), and Vortex Search (VS) of Dog and Ö lmez (2015) algorithms. In this method, similar to these algorithms, the search process is conducted in the specified boundaries. Contrary to these algorithms, that use different functions for decreasing the search space, in the present method the well-known GSA approach is employed to decrease the search boundaries. The minimization of an objective function is then performed by moving these search space into around the best global sample.

The present paper is organized as follows: In Section 2, we describe the well-known variance-based sensitivity approach. In Section 3, the new method is presented. A well-studied constrained optimization problem and four structural design examples are investigated in Section 4. Conclusions are derived in Section 5.

2. Variance-based sensitivity analysis

The search space boundary of the search algorithm proposed in this paper uses the variance-based SA theory, as the familiar method of GSA. As mentioned before, the purpose of SA is to measure the sensitivity of model inputs to output, referred to as sensitivity indicator (SI) of model inputs. In order to compute the SI of model inputs, suppose a model $Y=g(\boldsymbol{X})$, with $\boldsymbol{X}=[x_1,x_2,\ldots,x_n]$ being the model input vector, Y being the model output scalar, and $g()$ is a mapping function. The uncertainty of $\boldsymbol{X}$ propagates through $g(.)$ and results in the output model, Y. As the uncertainty of the output model is represented by its variance, $V(Y)$, to find the effect of an input $\boldsymbol{X_i}$ on the output, it is assumed that the true value of $\boldsymbol{X_i}$ can be determined by the variance reduction in the output, i.e., $V(Y)-V(Y|X_i=x_i^0)$, where x_i^0 is the true value of X_i and $V(Y|X_i=x_i^0)$ is the conditional expected value of V(Y), Since the true value is unknown, one can employ $V(Y)-E_{X_i}(V(Y|X_i))$ to evaluate the expected variance reduction in the output. The first order sensitivity indices of Y to the variable X_i can be expressed as (Zhai, Yang *et al*. 2014)

$$SI_i=\frac{V(Y)-V(Y\mid X_i=x_i^0)}{V(Y)} \tag{1}$$

In this equation, $V(Y)$ is the variance of output, Y. Also, $V(Y)-V(Y|X_i=x_i^0)$ is the variance reduction in the output for the particular value of $X_i=x_i^0$. Since we do not know what is the best value of X_i, one can measure $V(Y)-E_{X_i}(V(Y|X_i))$ to evaluate the expected variance reduction in the output. Therefore, the SI_i can be stated as

$$SI_i=\frac{V(Y)-E_{X_i}(V(Y\mid X_i)}{V(Y)}=1-\frac{E_{X_i}(V(Y\mid X_i)}{V(Y)} \tag{2}$$

In sensitivity analysis, SI_i varies between 0 and 1. The lower value of SI_i corresponds to the less influential X_i, the higher value of SI_i corresponds to the much influential X_i, and for $SI_i=0$, the X_i will have no influence on Y.

3. A global sensitivity analysis based algorithm

This section introduces a global sensitivity analysis based (GSAB) optimization algorithm, which is a single solution search method. The proposed algorithm is named as “a global sensitivity analysis (GSA)” because of determining the sensitivity indicator (*SI*) of design variables for guiding the search boundaries of the algorithm.

Meta-heuristic algorithms can be divided into two categories: population-based and single-solution, Kaveh and Mahdavi (2015), in the first group, a number of populations/agents are first generated and then all agents are updated iteratively until the termination condition is satisfied. In the other hand, single-solution meta-heuristics that are also known as trajectory methods, produce single solution by exploring the search space efficiently while reducing the effective size of the search space. The samples/populations of GSAB algorithm are used for two purposes: estimating the SI of design variables and single-solution of the algorithm. As these samples do not update iteratively, the proposed GSBA is studied within the single-solution meta-heuristic category. The feasibility space of samples in the GSAB algorithm updates for searching the optimal solution over several iterations. In each iteration, the feasibility space is updated using two values consisting of the sensitivity indicators and the global best sample. It is assumed that the problem is a minimization problem in R^D. The notations used are as follows:

S^t: The sample matrix in the *t*th iteration, $S^t = [X_i^t | i = 1, 2, \dots, N]$

X_i^t: The position of sample vector *i* in the *t*th iteration, $X_i^t = \{x_{ij}^t | j = 1,2,..,D\}$

X_{min}: The minimum allowable values vector of variables, $X_{min} = \{x_{min_j} | j = 1,2,..,D\}$

X_{max}: The maximum allowable values vector of variables, $X_{max} = \{x_{max_j} | j = 1,2,..,D\}$

$f(X_i)$: The fitness of vector *i*

UB^t: The upper search boundary vector of variables in the *t*th iteration, $UB^t = \{ub_j^t | j = 1,2,..,D\}$

LB^t: The lower search boundary vector of variables in the *t*th iteration, $LB^t = \{lb_j^t | j = 1,2,..,D\}$

BW^t: The band width of search space of variables in the *t*th iteration, $BW^t = \{bw_j^t | j = 1,2,..,D\}$

SF^t: The scale factor of band width of search space in the *t*th iteration, $SF^t = \{sf_j^t | j = 1,2,..,D\}$

Sbest: The global best sample (i.e. with lower fitness), $Sbest = \{sbest_j | j = 1,2,..,D\}$

R: A random vector within [0,1].

3.1 Methodology

The following steps outline the main procedure for the implementation of the GSAB.

Step 1. *Initialization:* The initial positions of samples are determined with random initialization in the search space

$$X_i^0 = X_{\min} + R(X_{\max} - X_{\min}), \quad i = 1,2,...,N \tag{3}$$

where X_i^0 determines the initial value vector of the *i*th sample; and *N* is the number of samples.

Step 2. *Calculating the sensitivity indices of variables*: In this step, the outputs (the objective function of optimization problem) are first calculated. The sensitivity analysis is performed next for the generated samples, and the sensitivity indicators (*SI*s) of variables are calculated.

The most well-known methods for calculating the variance-based sensitivity indicators are the Monte Carlo simulations; however these do not make full use of each output model evaluation. In order to calculate the variance-based sensitivity indicators from a given data, the scatterplot partitioning method can be utilized (Zhai, Yang *et al*. 2014), For this method, a single set of samples suffices to estimate all the sensitivity indicators. For estimating the variance-based sensitivity indices, suppose we have *N* points/samples $\{X^1, \dots, X^N\}$ and *N* model output samples $\{y^1, \dots, y^N\}$ obtained using the model $y=g(X)$, The variance of *Y* can be calculated by the sample variance $V(y)$, For the sample bounds of X_i as *[b1, b2]*, let it be decomposed into s successive, equal-probability and non-overlapping subintervals $A_k = [a_{k-1}, a_k)$, with $k=1, \dots, s$, $b_1 = a_0 < a_1 < \dots < a_k < \dots < a_s = b_2$ and $Pr(A_k)=1/s$. Decompose the output samples $\{y^1, \dots, y^M\}$ into s subsets according to the decomposition of X_i, where $B_k = \{y^j | x_i^j \in A_k\}, \quad k = 1,...,s$. The conditional variance $V(Y|x_i \in A_k)$ can then be evaluated by

$$V(Y|x_i \in A_k) = V(B_k) \tag{4}$$

The expected conditional variance, $E_{x_i}(V(Y|x_i))$, can now be approximately evaluated using

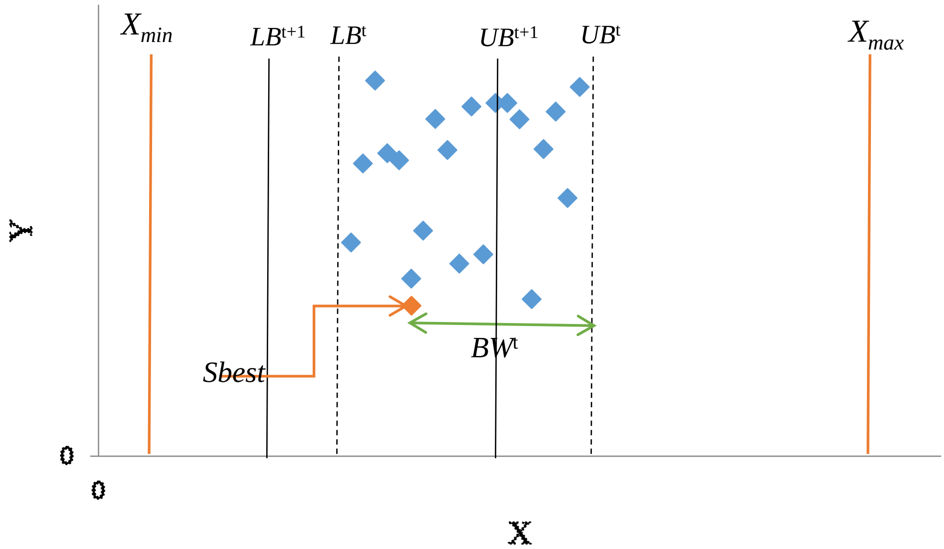


Fig. 1 An illustrative sketch of the search process

the following relationship

$$E_{x_i}(V(Y|x_i)) \approx \frac{1}{s}\sum_{k=1}^{s} V(B_k) \tag{5}$$

Ultimately, the sensitivity indicator of the *i*th variable, SI_i, is calculated using Eq. (2),

Step 3. *Defining the search boundaries:* In the GSAB algorithm, the search boundaries are moved around to the global best sample (which is updated and memorized in each iteration), *Sbest*, to push the samples into the feasible search space. The search boundaries are also decreased based on the values of sensitivity variables, which are evaluated in the previous step. Hence, the upper boundary and lower boundary of the search space of variables in the *t*+1th iteration can be computed by

$$\begin{aligned} UB^{t+1} &= Sbest + BW^{t} \times SF^{t} \leq X_{\max} \\ LB^{t+1} &= Sbest - BW^{t} \times SF^{t} \geq X_{\min} \end{aligned} \tag{6}$$

where BW^t and SF^t are the band width and scale factor of boundaries in the *t*th iteration, respectively (Fig. 1), Eq. (6) ensures that the current search space is moved around *Sbest* with the band width BW^t in the D-dimensional space. The vector BW^t can be calculated as

$$BW^{t} = \max(Sbest - LB^{t}, UB^{t} - Sbest) \tag{7}$$

For the algorithm to converge to a near-optimal solution, further exploitation (strong locality) is required to move the current solution towards to the optimal one. In the proposed GSAB algorithm, this is achieved by using a scale factor, *SF*. For this purpose, once *SI* values of variables are calculated, the most sensitive variable, i.e., variable with high *SI* value is identified for reducing the band width, and then the *SF* is calculated as

$$SF_j = \begin{cases} 1 - si_j & if \quad si_j = \max(SI) \\ 1 & Otherwisw \end{cases}, \quad \forall j = 1,\ldots,D \tag{8}$$

This equation shows that the band width of the most sensitive variable is decreased while other bands widths are constant in the *t*th iteration.

Step 4. *Replacement of the current samples:* In this step, the samples must be ensured to be inside the new search boundaries. For this purpose, the samples that exceed the boundaries are randomly regenerated in the new search boundaries (shown in Fig. 1) as

$$X_i^{t+1} = \begin{cases} X_i^t, & LB^{t+1} \le X_i^t \le UB^{t+1} \\ LB^{t+1} + R(UB^{t+1} - LB^{t+1}), & Otherwise \end{cases} \tag{9}$$

where i=1, 2, ..., N and t represents the iteration index.

Step 5. *Termination:* The optimization process is repeated from Step 2 until a termination criterion, such as maximum iteration number or no improvement of the best sample, is satisfied. In the GSAB algorithm, if the maximum band width of the search space, *max*(*W*), becomes smaller than 0.000001, the optimization process will be terminated. This is because the GSAB cannot change the search space of the agents. For the sake of clarity, the flowchart of the optimization procedure using the proposed GSAB is shown in Fig. 2.

4. Numerical examples

In this section, the efficiency of the proposed GSAB algorithm is shown through one

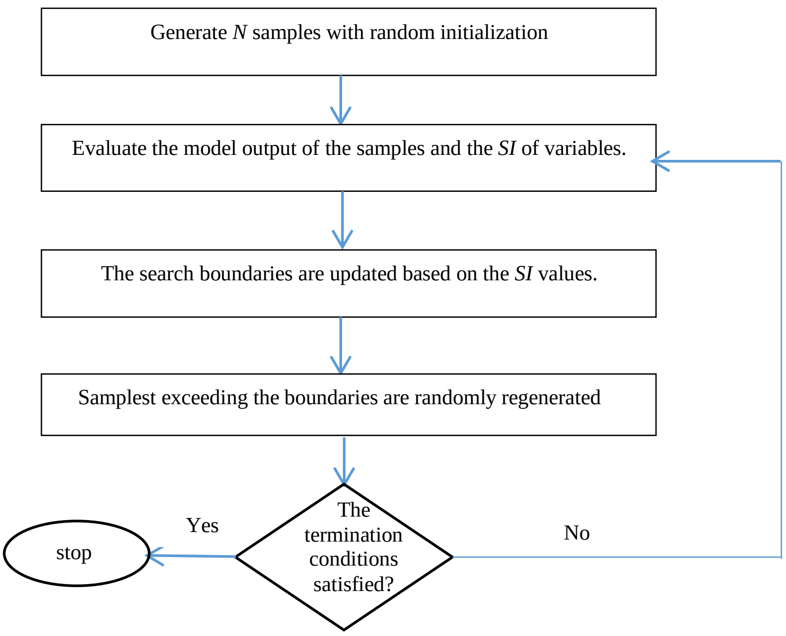


Fig. 2 The flowchart of the GSAB

mathematical constrained function and four well-studied truss structures under static loads taken from the optimization literature. These examples have been previously solved using a variety of other techniques, and are good examples to show the validity and effectiveness of the proposed algorithm. The first example shows the applicable of the GSAB for optimization of the constrained problems. In Example 2, a planar truss structure is studied for finding the optimal cross sections. Examples 3, 4 and 5 are selected to show the importance of selection of optimization algorithm in reducing the number of function evaluations.

In structural optimization problems the main objective is to minimize the weight of the structures under some constraints. The optimization problem for a truss structure can be stated as follows

$$\begin{array}{ll} \text{Find} & X = [x_1, x_2, x_3, .., x_n] \\ \text{to minimizes} & W(X) = \sum_{i=1}^{ne} \rho_i A_i l_i \\ \text{subjected to} & g_j(X) \le 0,\ j = 1,2,\ldots,m \\ & x_{lmin} \le x_l \le x_{lmax} \end{array} \tag{10}$$

where X is the vector of all design variables with n unknowns; W is the weight of truss structure; ρ_i, A_i and l_i are mass density, cross sectional area and length of ith member, respectively; ne is number of the structural elements; g_j is the jth constraint from m inequality constraints. Also, x_{lmin} and x_{lmax} are the lower and upper bounds of design variable vector, respectively.

The employed constraint handling is the penalty function approach proposed by Deb (2000), It should be noted that the output model of SA method is the penalized objective function. For 2nd through 4th and engineering design examples, the numbers of N=40 and N=20 samples are utilized, respectively. Also, these examples are independently optimized 20 times. In the final example, N=50 samples are considered. The algorithm is coded in MATLAB. Structural analysis is performed by the direct stiffness method.

4.1 Design of a tension/compression spring

This problem was first described by Belegundu (1982) and Arora (1989), It consists of minimizing the weight of a tension/compression spring subject to constraints on shear stress, surge frequency, and minimum deflection as shown in Fig. 3. The design variables are the mean coil diameter $D(= x_1)$; the wire diameter $d(= x_2)$, and the number of active coils $N(= x_3)$, The problem can be stated as follows

$$\text{Find} \quad \{x_1, x_2, x_3\} \tag{11}$$

To minimize

$$\cos t(x) = (x_3 + 2)x_2 x_1^2 \tag{12}$$

Subject to

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0$$
$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0 \tag{13}$$

The bounds on the design variables are

$$0.05 \le x_1 \le 2, \quad 0.25 \le x_2 \le 1.3, \quad 2 \le x_3 \le 15, \tag{14}$$

This problem has been solved by Belegundu (1982) using eight different mathematical optimization techniques. Arora (1989) solved this problem using a numerical optimization technique called a constraint correction at the constant cost. Coello (2000) as well as Coello and Montes (2002) solved this problem using GA-based method. Additionally, He and Wang (2007) utilized a co-evolutionary particle swarm optimization (CPSO), Recently, Montes and Coello (2008), Kaveh and Talatahari (2010), Kaveh and Mahdavi (2014) used the ES, CSS and CBO to solve this problem, respectively.

Tables 1 and 2 compare the best results obtained in this paper and those of the other researches. The GSAB found the best cost as 0.0126652 after 3,729 fitness function evaluations. Although, the

Table 1 Comparison of GSAB optimized designs with literature for the tension/compression spring problem

Methods	Optimal design variables			f(x)
	x_1 (d)	x_2 (D)	x_3 (N)	
Belegundu (1982)	0.050000	0.315900	14.250000	0.0128334
Arora (1989)	0.053396	0.399180	9.185400	0.0127303
Coello (2000)	0.051480	0.351661	11.632201	0.0127048
Coello and Montes (2002)	0.051989	0.363965	10.890522	0.0126810
He and Wang (2007)	0.051728	0.357644	11.244543	0.0126747
Montes and Coello (2008)	0.051643	0.355360	11.397926	0.012698
Kaveh and Talatahari (2010)	0.051744	0.358532	11.165704	0.0126384
Kaveh and Mahdavi (2014)	0.051894	0.3616740	11.007846	0.0126697
Present work	**0.05171604**	**0.3573671**	**11.2509979**	**0.0126652**

Table 2 Statistical results from different optimization methods for tension/compression string problem

Methods	Best result	Average optimized cost	Worst result	Std Dev	Fitness function evaluations
Belegundu (1982)	0.0128334	N/A	N/A	N/A	N/A
Arora (1989)	0.0127303	N/A	N/A	N/A	N/A
Coello (2000)	0.0127048	0.012769	0.012822	3.9390e-5	900,000
Coello and Montes (2002)	0.0126810	0.0127420	0.012973	5.9000e-5	N/A
He and Wang (2007)	0.0126747	0.012730	0.012924	5.1985e-5	200,000
Montes and Coello (2008)	0.012698	0.013461	0.16485	9.6600e-4	25,000
Kaveh and Talatahari (2010)	0.0126384	0.012852	0.013626	8.3564e-5	4,000
Kaveh and Mahdavi (2014)	0.0126697	0.01272964	0.0128808	5.00376e-5	4,000
Present work	0.0126652	0.012875334	0.01334400	2.31935e-4	3,729

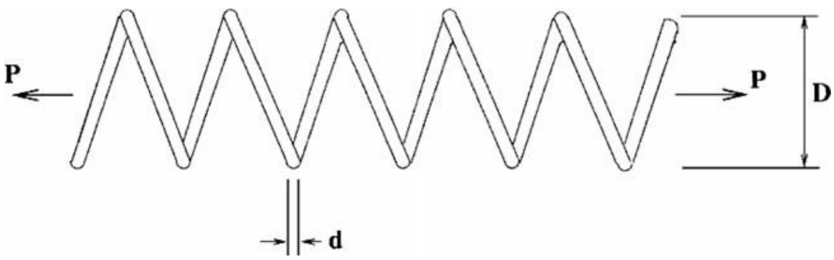
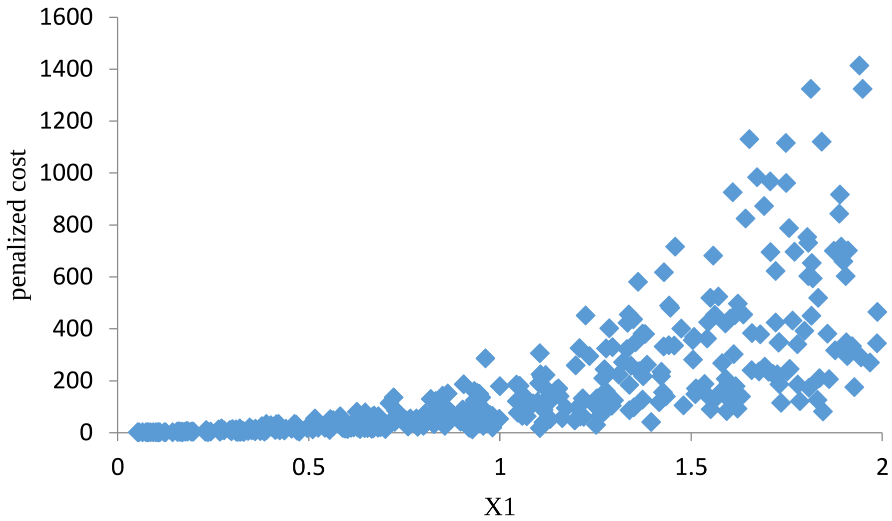
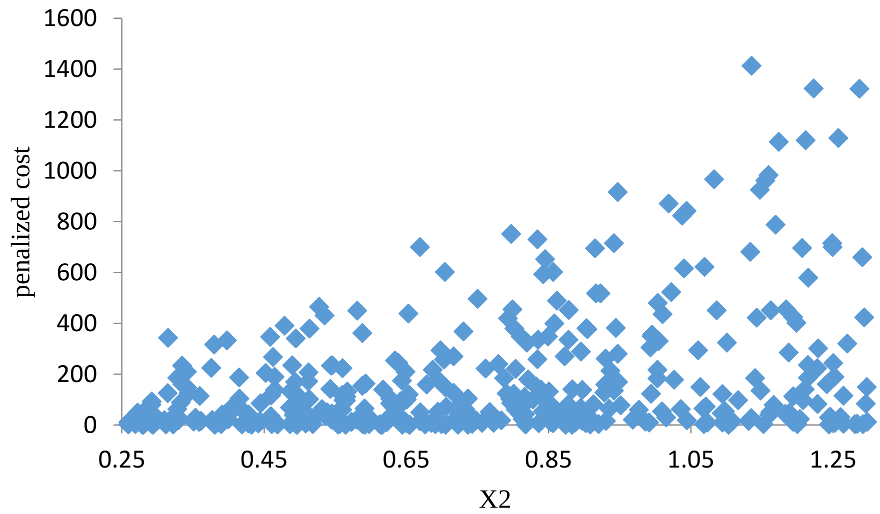


Fig. 3 Schematic of the tension/compression spring with indication of design variables

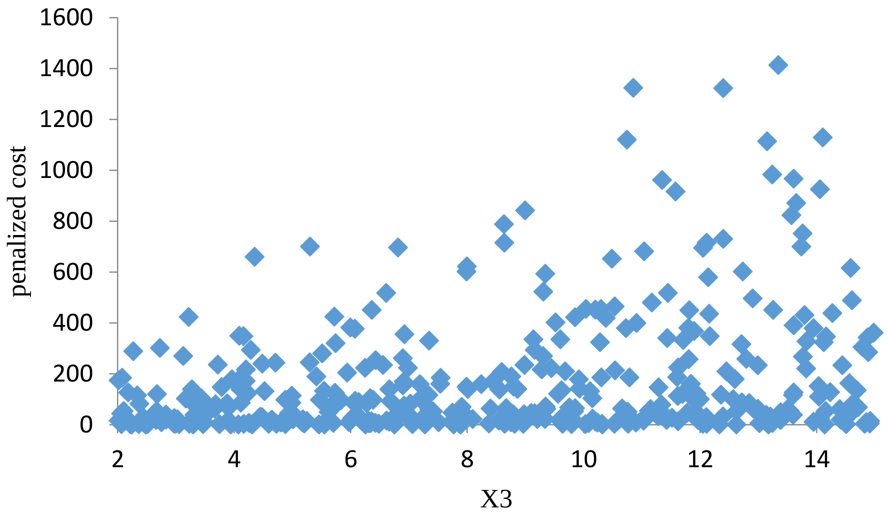


(a)



(b)

Fig. 4 Scatter plots for variables: (a) X_1; (b) X_2; (c) X_3 of the first example



(c)

Fig. 4 Continued

best cost found was more than the standard CSS, it is the lowest fitness function evalutions amongst the existing literature results. It should be noted that the lighter design found by Kaveh and Talatahari (2010) slightly violates the first two optimization constraints.

In order to show the performance of the GSA method in the GSAB algorithm, a study is focused on the influence of the *SI*s on the proposed algorithm result. As described in Section 3.1, the GSA method requires two pre-defined parameters: the number of samples, N, and the number of subintervals, s. A larger number of samples leads to an increase of the accuracy of the sensitivity indicators. In the other hand, because of generating the output model of the GSA method, the fitness function (or output) evaluations increases with the number of samples. The number of subintervals can also be affected to the SI values. As Zhai, Yang *et al*. (2014) underlines, the appropriate number of subintervals can be considered as $s = \frac{N}{5}$. The scatterplots of $X_{i=1,2,3}$ and *cost* for N=100 samples are shown in Fig. 4. It can be noticed that: (i) x_1 seems to be the most influential input; (ii) x_2 and x_3 seem to be the low influential inputs, because the distribution of samples against the first variable, x_1, is denser compared to other variables. This is confirmed by the GSA method. If we apply the space-partition variance-based sensitivity analysis approach, we obtain the sensitivity indicators, *SI*s, as shown in the Fig. 5. It can be seen from this figure that the *SI* of the first variable is higher than other variables, i.e. the most influential/sensitive variable is the first variable.

Fig. 6 shows the convergence rates of the upper and lower boundary of the search space and best ones in the optimization process. As previously mentioned, here the number of samples is considered as N=40 in the optimization process. It can be seen, with respect to the second and third variables, that the search space of the first variable is rapidly decreased in the early iterations because it is more sensitive to the output (i.e., objective function), Hence, despite the fewer samples, the proposed GSA approach could appropriately rank the variables based on these sensitivities.

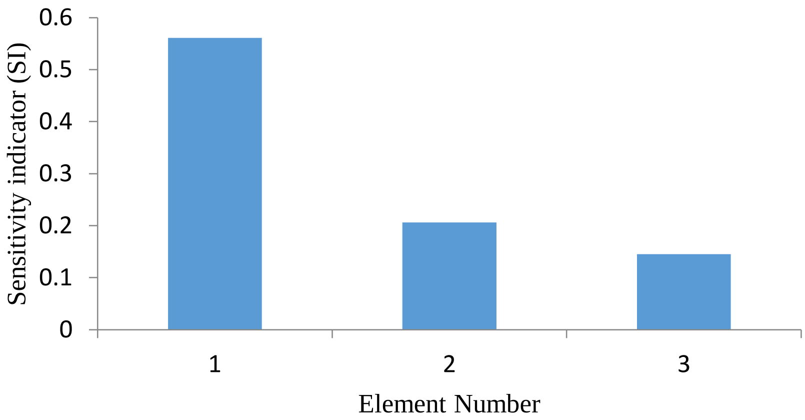
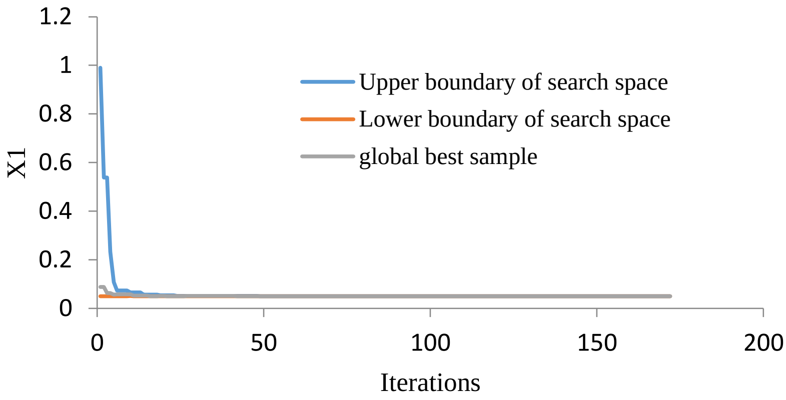
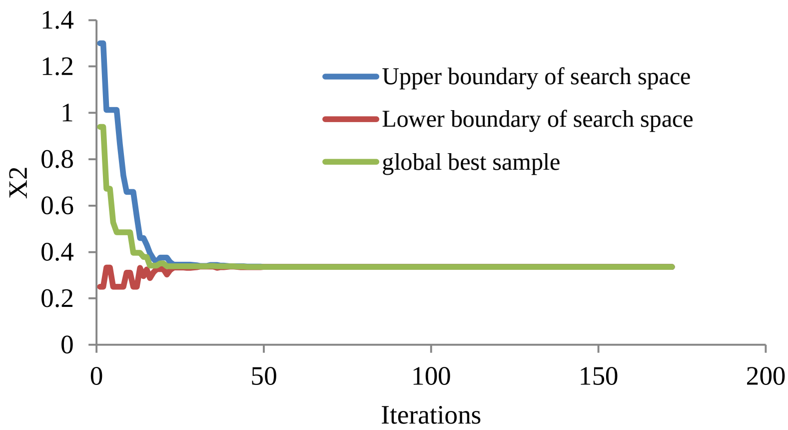


Fig. 5 The obtained sensitivity indicator of variables for penalized cost function of the first example

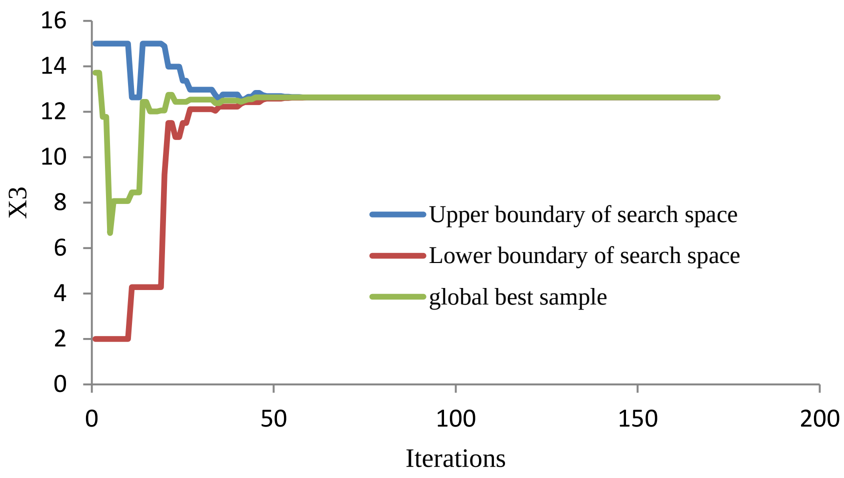


(a)



(b)

Fig. 6 The convergence history graphs of search space for variables: (a) X_1; (b) X_2; (c) X_3



(c)

Fig. 6 Continued

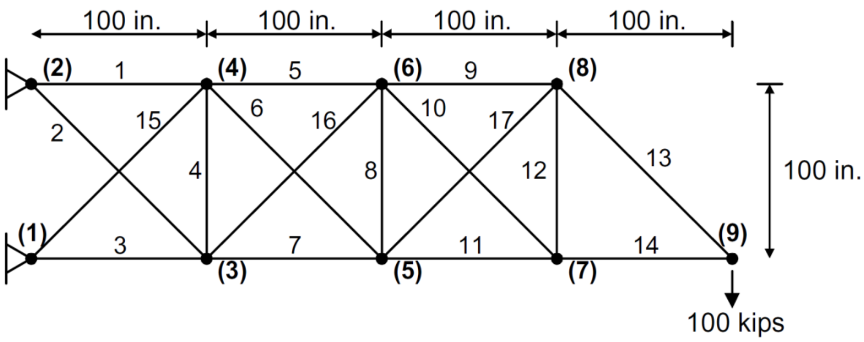


Fig. 7 Schematic of the planar 17-bar truss problem

The optimum variables found with different algorithms can also be used for comparing the *SI* of variables. As shown in Table 1, although the optimal objective functions found by different optimization algorithms have not significant difference, but the values of the optimum second and third variables have significant difference compared to the first variable.

4.2 Planar 17-bar truss problem

A 17-bar planar truss is schematized in Fig. 7. The single vertical downward load of 100 kips at node 9 is considered and there are 17 independent design variables. The elastic modulus is 30,000 ksi and the material density is 0.268 lb/in^3 for all elements. The members are subjected to the stress limits of 50 ksi both in tension and compression. Displacement limitations of ±2.0 in are imposed on all nodes in both directions (*x* and *y*), The allowable minimum cross-sectional area of all the elements is set to 0.1 in^2.

Table 3 Comparison of the optimized designs for the 17-bar planar truss

Element Group	Optimal cross-sectional areas				
	Khot and Berke (1984)	Adeli and Kumar (1995)	Kaveh and Ilchi (2014)		Present work
			CBO	ECBO	
A_1	15.930	16.029	15.9674	15.9158	15.8916
A_2	0.100	0.107	0.1386	0.1001	0.10088
A_3	12.070	12.183	12.1735	12.0762	12.00129
A_4	0.100	0.110	0.1000	0.1000	0.100015
A_5	8.067	8.417	7.8524	8.0527	8.078015
A_6	5.562	5.715	5.5447	5.5611	5.571161
A_7	11.933	11.331	11.9648	11.9470	11.98603
A_8	0.100	0.105	0.1002	0.1000	0.100602
A_9	7.945	7.301	7.9385	7.9425	8.009118
A_{10}	0.100	0.115	0.1003	0.1000	0.100585
A_{11}	4.055	4.046	4.1146	4.0589	4.06476
A_{12}	0.100	0.101	0.1000	0.1000	0.100046
A_{13}	5.657	5.611	5.8134	5.6644	5.577003
A_{14}	4.000	4.046	4.0556	4.0057	4.004148
A_{15}	5.558	5.152	5.4973	5.5565	5.611166
A_{16}	0.100	0.107	0.1329	0.1000	0.104159
A_{17}	5.579	5.286	5.4043	5.5740	5.568715
Best weight (lb)	2581.89	2594.42	2582.79	2581.89	2582.032
Average weight (lb)	N/A	N/A	2631.07	2597.11	2585.62
Standard deviation (lb)	N/A	N/A	49.45	22.43	9.248879

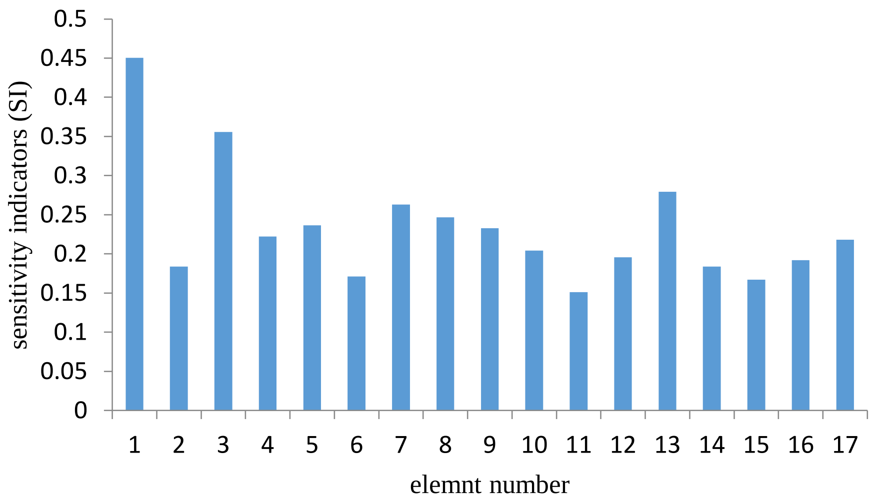
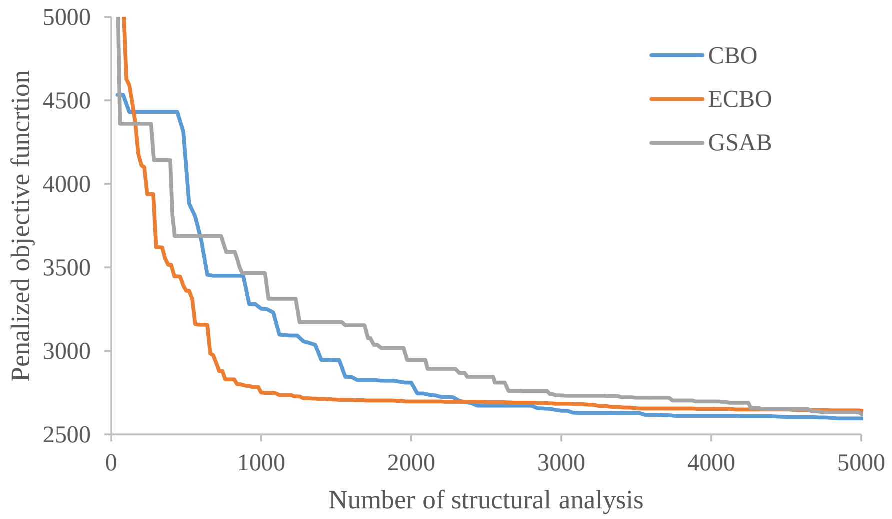
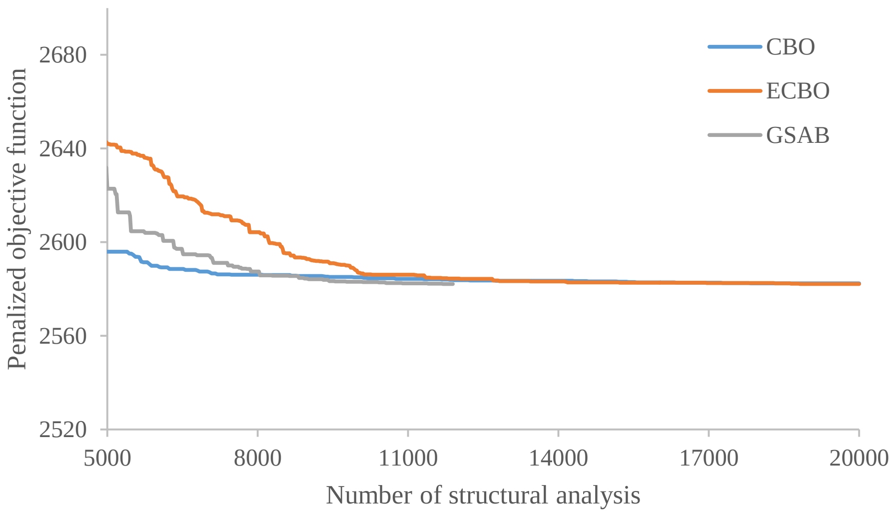


Fig. 8 The obtained sensitivity indicator of variables for the penalized weight of the planar 17-bar truss problem



(a)



(b)

Fig. 9 Convergence curves obtained for the 17-bar truss problem for: (a) 0-500th; (b) 500-2000th number of analyses

Table 3 presents the optimum designs obtained by Khot and Berke (1984), Adeli and Kumar (1995), standard CBO, ECBO, Kaveh and Ilchi Ghazaan (2014) and the proposed GSA algorithms. Although, the best design is obtained by the ECBO and the work of Khot and Berke (1984), the average weight and standard deviation of independent runs obtained by the GSAB are the lowest. The optimization process of the best run of the GSAB is completed in 12,255 analyses. Standard CBO and ECBO required 15,560 and 14,180 analyses to converge to the optimum. The SI values of variables are shown in Fig. 8. It can be seen from the Fig. 8 and Table 3 that the sensitivity of members 1,3,5,7,9,11 and 13 are more than the remaining members, and the larger optimum designs obtained using the optimization algorithms have the high value of *SI*s. Convergence curves of the GSAB, ECBO and CBO are shown in Fig. 9. Although CBO and ECBO were considerably faster in the early optimization iterations, GSAB converged to a significantly better design without being trapped in local optima.

4.3 A 72-bar spatial truss structure

Schematic topology and element numbering of a 72-bar space truss is shown in Fig. 10. The elements are classified into 16 design groups according to Table 4. The material density is 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is taken as 10,000 ksi (68,950 MPa), The members are subjected to the stress limits of ±25 ksi (±172.375 MPa), The uppermost nodes are subjected to the displacement limits of ±0.25 in (±0.635 cm) in both x and y directions. The minimum permitted cross-sectional area of each member is taken as 0.10 in^2 (0.6452 cm^2), and the maximum cross-sectional area of each member is 4.00 in^2 (25.81 cm^2), The loading conditions are considered as:

1. Loads 5, 5 and −5 kips in the *x, y and z* directions at node 17, respectively;
2. A load equal to −5 kips in the *z* direction at nodes 17, 18, 19 and 20.

Table 4 shows the optimum design variables using the GSAB algorithm, which is compared to results of the other algorithms. The best result of the GSAB approach is 379.7689, while this is 385.76, 380.24, 381.91, 379.85, 380.458, 379.75 and 379.77 *Ib* for the GA Erbatur, Hasançebi *et*

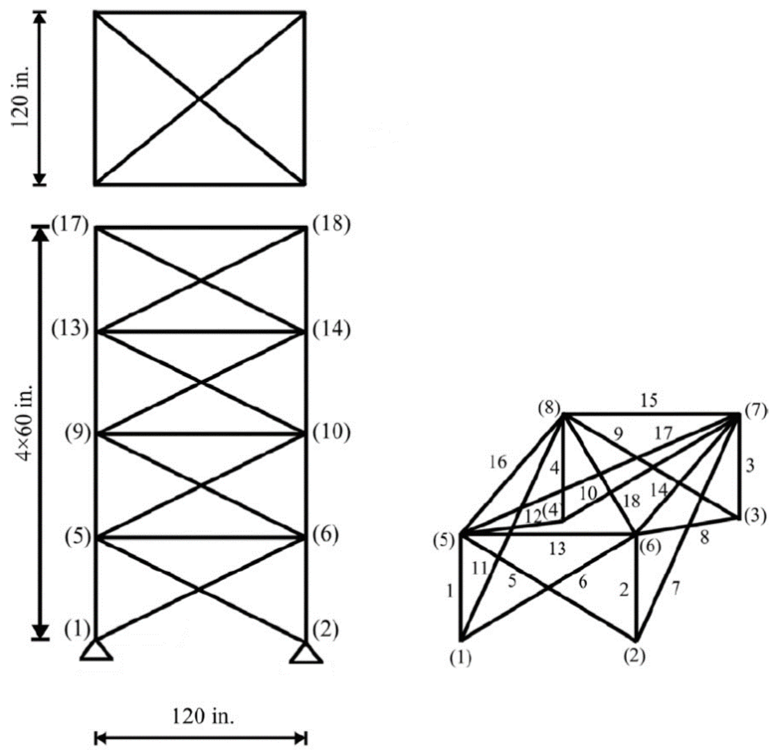


Fig. 10 Schematic of a 72-bar spatial truss

Table 4 Comparison of GSAB optimized designs with those of literature for the 72-bar spatial truss (in^2)

Element group	Optimal cross-sectional areas (in^2)							
	Erbatur *et al*. (2014) GA	Camp *et al*. (2004) ACO	Perez *et al*. (2007) PSO	Camp (2007) BB-BC	Kaveh *et al*. (2013) RO	Kaveh and Ilchi. (2014) CBO	Kaveh and Ilchi. (2014) ECBO	Present work
1-4	1.755	1.948	1.7427	1.8577	1.8365	1.9170	1.8519	1.909083519
5-12	0.505	0.508	0.5185	0.5059	0.5021	0.5031	0.5141	0.515793736
13-16	0.105	0.101	0.1000	0.1000	0.1000	0.1000	0.1000	0.100097411
17-18	0.155	0.102	0.1000	0.1000	0.1004	0.1001	0.1000	0.100154463
19-22	1.155	1.303	1.3079	1.2476	1.2522	1.2721	1.2819	1.292073465
23-30	0.585	0.511	0.5193	0.5269	0.5033	0.5050	0.5091	0.524173699
31-34	0.100	0.101	0.1000	0.1000	0.1002	0.1000	0.1000	0.100059742
35-36	0.100	0.100	0.1000	0.1012	0.1001	0.1000	0.1000	0.100103578
37-40	0.460	0.561	0.5142	0.5209	0.5730	0.5184	0.5312	0.515818559
41-48	0.530	0.492	0.5464	0.5172	0.5499	0.5362	0.5173	0.513756471
49-52	0.120	0.1	0.1000	0.1004	0.1004	0.1000	0.1000	0.100010199
53-54	0.165	0.107	0.1095	0.1005	0.1001	0.1000	0.1000	0.100509039
55-58	0.155	0.156	0.1615	0.1565	0.1576	0.1569	0.1560	0.157384016
59-66	0.535	0.550	0.5092	0.5507	0.5222	0.5374	0.5572	0.526496976
67-70	0.480	0.390	0.4967	0.3922	0.4356	0.4062	0.4259	0.407510273
71-72	0.520	0.592	0.5619	0.5922	0.5971	0.5741	0.5271	0.56965198
Best Weight (Ib)	385.76	380.24	381.91	379.85	380.458	379.75	379.77	379.7689
Average Weight (Ib)	N/A	383.16	N/A	382.08	382.553	380.03	380.39	380.3613
Std dev	N/A	3.66	N/A	1.912	1.221	0.278	0.8099	0.519822
No. of analyses	N/A	18,500	N/A	19,621	19,084	16,000	18,000	13,795

al. (2014), ACO Camp and Bichon (2004), PSO Perez and Behdinan (2007), BB-BC Camp (2007), RO Kaveh, Ilchi Ghazaan *et al*. (2013), CBO and ECBO, Kaveh and Ilchi Ghazaan (2014) algorithms, respectively. Also, the number of analyses of the GSAB is 13795, while it is 18500, 19621, 19084, 16000 and 18000 for the ACO, BB-BC, RO, CBO and ECBO algorithms, respectively. It is evident from Table 4 that although the statistical results of 20 independent runs for the CBO is less than the GSAB algorithm, the number of functions evaluation for the GSAB algorithm is less than that of the CBO. Fig. 11 shows the *SI* values of the variables for this example. Fig. 11 shows the maximum stress ratios in truss group members obtained using the GSAB. As can be seen from Figs. 11 and 12, and Table 4, the first design variable, i.e. the first story columns area, is the most sensitive variable because of the high amount of axial force in the first story columns. The design variables corresponding to the vertical braces area are also the sensitive variables, with respect to other truss group members, because these can affect the displacement constraints and can have high length in the shape of truss. Fig. 13 shows the convergence curves of the GSAB, ECBO and CBO obtained for this test case.

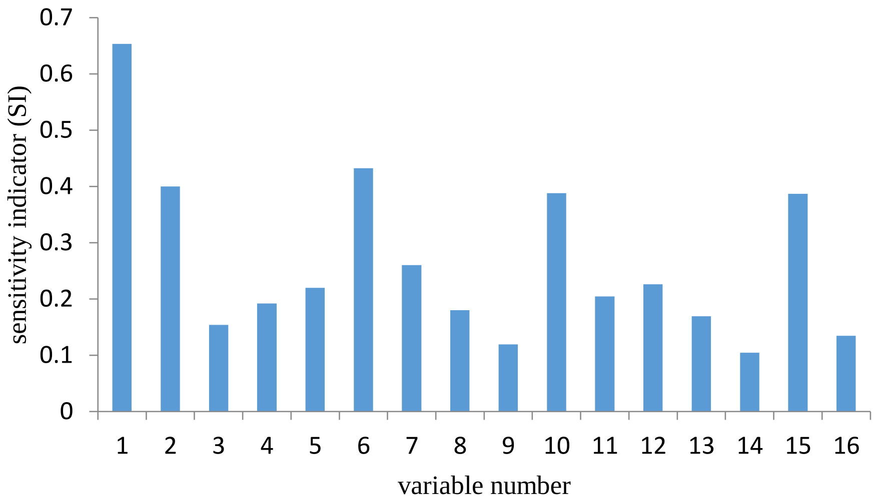


Fig. 11 The obtained sensitivity indicator of variables for the penalized weight of 72-bar spatial truss

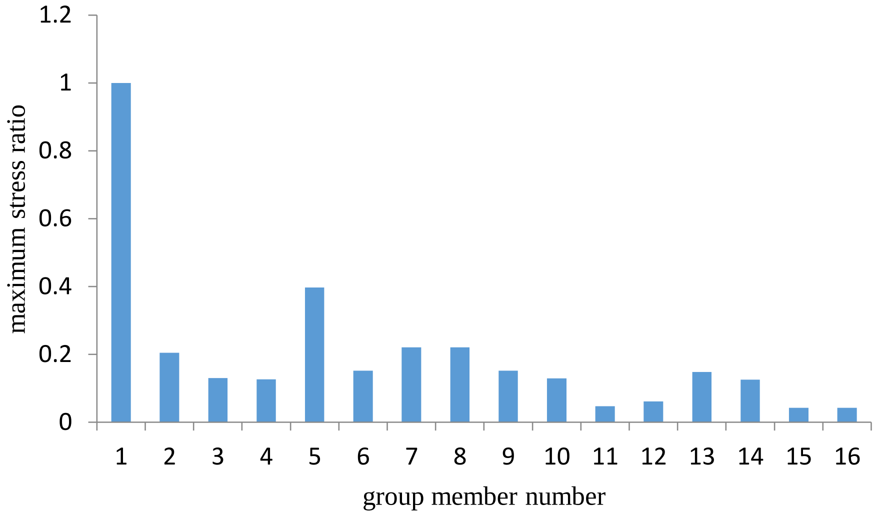
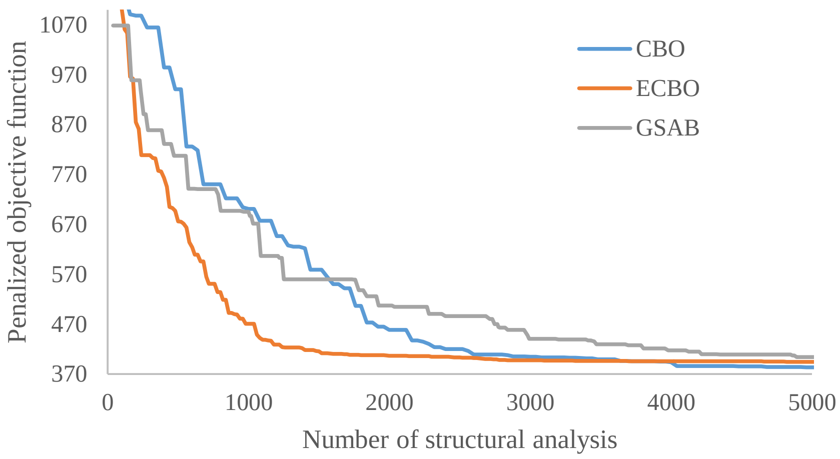


Fig. 12 The maximum stress ration in the group elements of the 72-bar truss structure

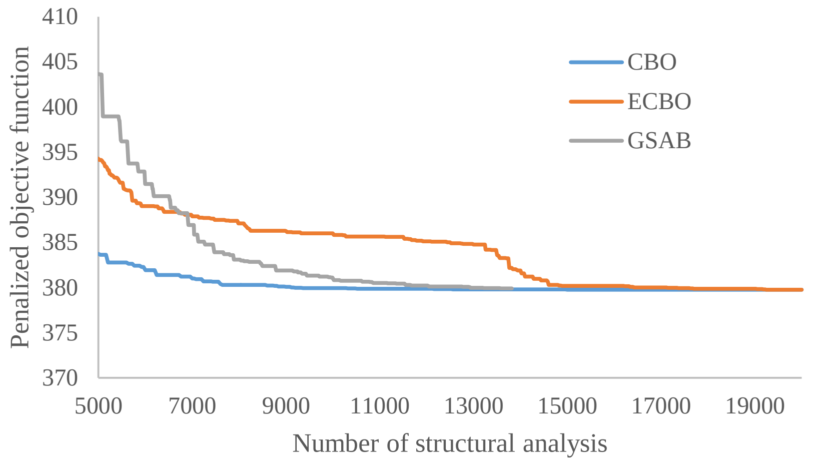
4.4 A 120-bar truss dome

The fourth case solved in this study is the weight minimization problem of the 120-bar truss dome shown in Fig. 14. This test case was investigated by Soh and Yang (1996) as a configuration optimization problem. It has been solved later as a sizing optimization problem by Kaveh and Talatahari (2010), Kaveh and Khayatazad (2012) and Kaveh and Mahdavi (2014),

The allowable tensile and compressive stresses are set according to the AISC ASD (1989) code, as follows



(a)



(b)

Fig. 13 Convergence curves obtained in the 72-bar truss problem for: (a) 0-5000th; (b) 5000-20000th number of analyses

$$\begin{cases} \sigma_i^+ = 0.6F_y & for \sigma_i \geq 0 \\ \sigma_i^- & for \sigma_i \leq 0 \end{cases} \tag{15}$$

where σ_i^- is calculated according to the slenderness ratio

$$\sigma_i^- = \begin{cases} \left[\left(1 - \frac{\lambda_i^2}{2C_c^2}\right)F_y\right] \Big/ \left(\frac{5}{3} + \frac{3\lambda_i}{8C_c} - \frac{\lambda_i^3}{8C_c^3}\right) & for \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & for \lambda_i \geq C_c \end{cases} \tag{16}$$

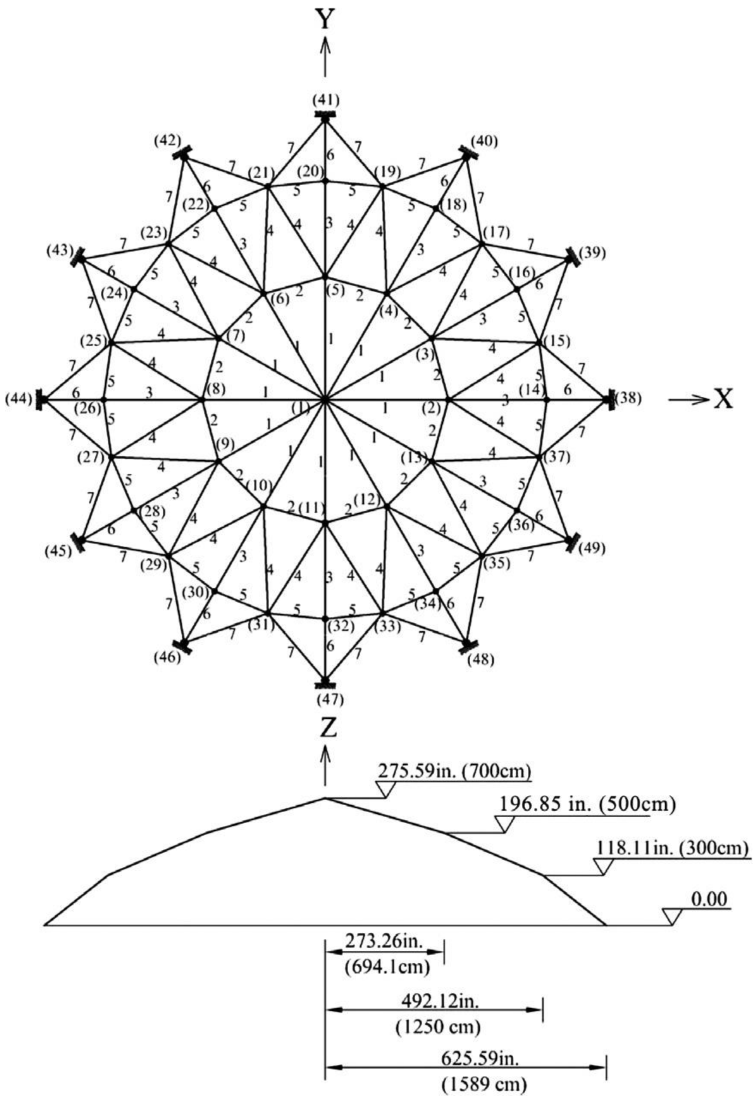


Fig. 14 Schematic of the spatial 120-bar dome truss with indication of design variables and main geometric dimensions

where E is the modulus of elasticity, F_y is the yield stress of steel, C_c is the slenderness ratio (λ_i) dividing the elastic and inelastic buckling regions ($C_c = \sqrt{2\pi^2 E / F_y}$),λ_i is the slenderness ratio ($\lambda_i = \frac{KL_i}{r_i}$), K is the effective length factor, L_i is the member length and r_i is the radius of gyration.

The modulus of elasticity is 30,450 ksi and the material density is 0.288 lb/in^3. The yield stress of steel is taken as 58.0 ksi. On the other hand, the radius of gyration (r_i) is expressed in terms of cross-sectional areas as $r_i = aA_i^{\,b}$ (Saka 1990), Here, a and b are constants depending on the types of sections adopted for the members such as pipes, angles, and tees. In this example, pipe sections

Table 5 Comparison of the GSAB optimized designs with those of literature for the 120-bar dome problem

Element group	Optimal cross-sectional areas (in^2)					
	Kaveh *et al.* (2013) PSO	Kaveh *et al.* (2013) PSOPC	Kaveh *et al.* (20103) HPSACO	Kaveh *et al.* (2013) RO	Kaveh and Ilchi. (2014) CBO	Present work
1	12.802	3.040	3.095	3.030	3.0284	3.024214
2	11.765	13.149	14.405	14.806	14.9543	14.8525
3	5.654	5.646	5.020	5.440	5.4607	5.064194
4	6.333	3.143	3.352	3.124	3.1214	3.134918
5	6.963	8.759	8.631	8.021	8.0552	8.457656
6	6.492	3.758	3.432	3.614	3.3735	3.283562
7	4.988	2.502	2.499	2.487	2.4899	2.49657
Best weight (Ib)	**51986.2**	**33481.2**	**33248.9**	**33317.8**	**33286.3**	**33249.68**
Average weight (Ib)	**-**	**-**	**-**	**-**	**33398.5**	**33253.32**
Std (Ib)	**-**	**-**	**-**	**354.333**	**67.09**	**4.112399**

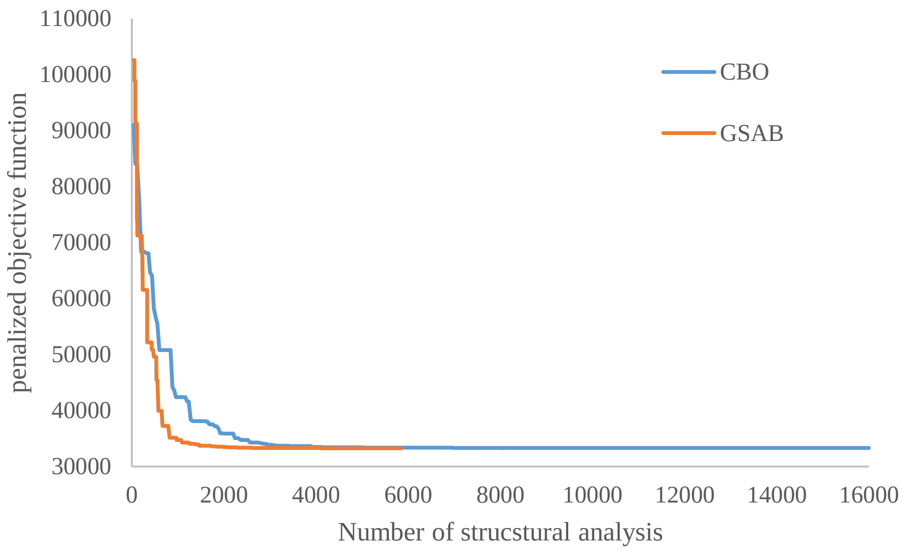
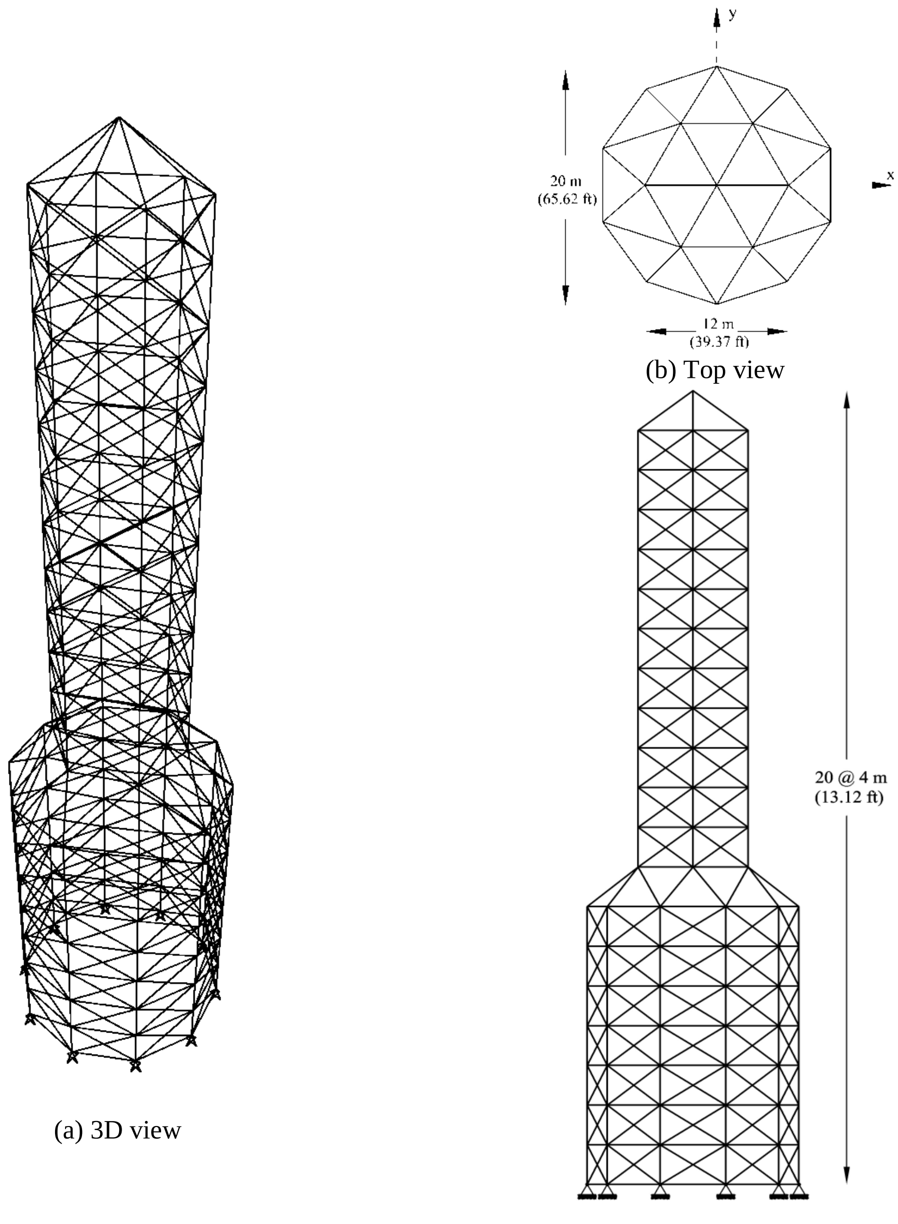


Fig. 15 The convergence curves for the 120-bar dome truss

(a=0.4993 and b=0.6777) are adopted for bars. All members of the dome are divided into seven groups, as shown in Fig. 14. The dome is considered to be subjected to vertical loads at all the unsupported joints. These are taken as −13.49 kips (60 kN) at node 1, −6.744 kips (30 kN) at nodes 2 through 14, and −2.248 kips (10 kN) at the remaining of the nodes. The minimum cross-sectional area of elements is 0.775 in^2 (cm^2), In this example, the constraints are considered: stress constraints and displacement limitations of ±0.1969 in imposed on all nodes in all directions. The maximum cross-sectional area is also considered as 20.0 in^2.

Table 5 summarizes the results obtained by the present work and those of the previously



(a) 3D view

(b) Top view

(c) Side view

Fig. 16 Schematic of the spatial 582-bar tower

reported researches. As it can be seen, the best results obtained using the GSAB is better than those of the other methods (except for the HPSACO), The standard deviations of results are also better than the RO and CBO algorithms. In this example, the GSAB needs 5,823 analyses to find the optimum result while this number is 10,000, 125,000, 19,800 and 16,000 for the HPSACO, PSOPC, RO and CBO algorithms as reported, respectively. Convergence curves of the GSAB, ECBO and CBO are shown in Fig. 15.

4.5 A 582-bar tower truss

The 582-bar spatial truss structure, shown in Fig. 16, was studied with discrete variables by other researchers (Kaveh and Mahdavi 2014b, Hasançebi, Çarbas *et al*. 2009), However, this structure recently have been used with continuous sizing variables by (Kaveh and Mahdavi 2014c), The 582 structural members are categorized as 32 independent size variables. A single load case is considered consisting of lateral loads of 5.0 kN (1.12 kips) applied in both x- and y-directions and a vertical load of −30 kN (−6.74 kips) applied in the z-direction at all nodes of the tower. The lower and upper bounds on size variables are taken as 3.1 in^2 (20 cm^2) and 155.0 in^2 (1000 cm^2), respectively.

The stress constraint is applied to this problem similar to the previous example. The maximum slenderness ratio is also limited to 300 for tension members, and it is recommended to be limited to 200 for compression members according to ASD-AISC (1989), The modulus of elasticity is 29,000 ksi (203893.6 MPa) and the yield stress of steel is taken as 36 ksi (253.1 MPa), Other constraints are the limitations of nodal displacements which should be no more than 8.0 cm (3.15 in.) in all directions.

Table 6 lists the optimal values of the 32 size variables obtained by the CBO and present algorithm. Fig. 17 shows the convergence diagrams for both algorithms. The numbers of structural analyses are achieved as 20,000 and 17,127 using the CBO and the presented algorithm, respectively. It is evident that the GSAB is better than CBO in term of best weight of the results, numbers of structural analyses and convergence rate.

Table 6 Optimum design cross-sections obtained for the 582-bar tower truss

Element groups	CBO (Kaveh and Mahdavi 2014b)	Present Work	Element groups	CBO (Kaveh and Mahdavi 2014b)	Present Work
	Area, cm^2			Area, cm^2	
1	20.5526	20.55701	17	155.6601	136.2476
2	162.7709	164.7262	18	21.4951	24.73753
3	24.8562	22.94746	19	25.1163	20.31951
4	122.7462	149.5989	20	94.0228	102.9622
5	21.6756	20.11998	21	20.8041	20.95159
6	21.4751	21.31396	22	21.223	20.2722
7	110.8568	104.9726	23	53.5946	67.30363
8	20.9355	21.32158	24	20.628	20.01284
9	23.1792	20.30167	25	21.5057	20.34263
10	109.6085	124.8855	26	26.2735	22.69424
11	21.2932	20.62465	27	20.6069	20.32811
12	156.2254	161.9005	28	21.5076	20.85621
13	159.3948	150.507	29	24.1394	22.46429
14	107.3678	112.6909	30	20.2735	21.85471
15	171.915	151.3205	31	21.1888	23.34649
16	31.5471	29.25933	32	29.6669	21.70401
Volume (m^3)				16.1520	16.0877

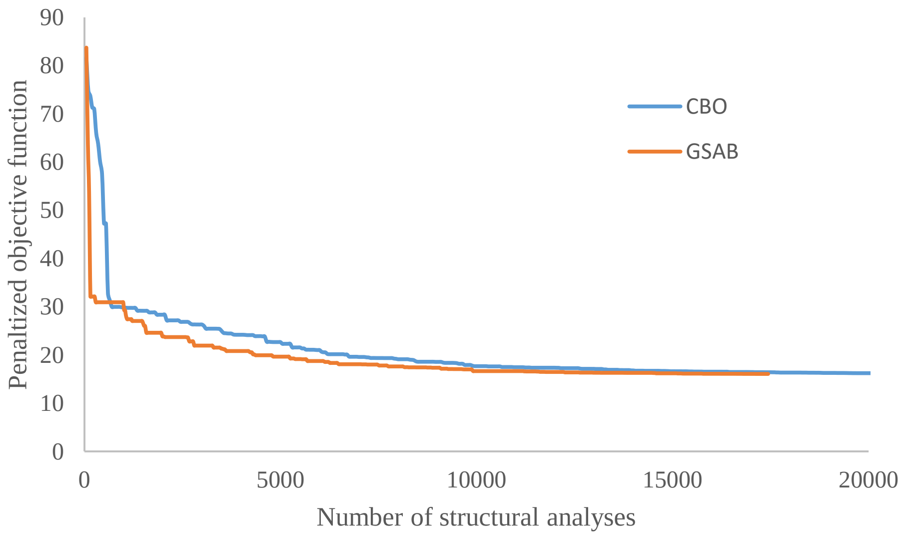


Fig. 17 The convergence curves of the CBO and GSAB for 582-bar tower truss

5. Conclusions

In this paper, a new single-solution global sensitivity analysis based optimizer called GSAB is developed. Compared to other meta-heuristic algorithms, the GSAB has several distinct features. Firstly, the population/agents in GSAB are directly represented by the samples, which are used to find the sensitivity values of the design variables as well as the optimization search in sequence at each iteration. Hence, one can consider the proposed algorithm as a single-solution meta-heuristic category. Secondly, the search boundaries are considered and these are decreased based on the sensitivity values of the variables at each iteration. The sample, which is found as the best one, is also selected to push the search boundaries around this sample and it is selected as solution of the GSAB algorithm. Then, the samples that exceed the search boundaries are randomly regenerated in the boundaries. Unlike the common meta-heuristic algorithms where the agent of a population move to the new positions without considering any information about the sensitivity of variables, in this algorithm the search boundaries are decreased based on the sensitivity indices of the variables, and this accelerates the converge of the solution.

The GSAB algorithm is tested over five benchmark optimization problems consisting of mathematical and truss structure optimization problems with different dimensions. The results are compared to those of some population based meta-heuristics. This comparison reveals that besides its simplicity, the proposed GSAB algorithm is also competitive, especially from the number of functions evaluation point of view, when compared to the performance of the some other algorithms.

Acknowledgements

The first author is grateful to the Iran National Science Foundation for the support.

References

Adeli, H. and Kumar, S. (1995), "Distributed genetic algorithm for structural optimization", *J. Aerospace Eng.*, ASCE, **8**(3), 156-163.

American Institute of Steel Construction (AISC) (1989), Manual of Steel Construction Allowable Stress Design, 9th Edition, Chicago, IL, USA.

Archer, G., Saltelli, A. and Sobol, I. (1997), "Sensitivity measures, ANOVA-like techniques and the use of bootstrap", *J. Statist. Comput. Simul.*, **58**, 99-120.

Arora, J.S. (1989), *Introduction to Optimum Design*, McGraw-Hill, New York, USA.

Belegundu, A.D. (1982), "A study of mathematical programming methods for structural optimization", Ph.D. Thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa, USA.

Camp, C.V. (2007), "Design of space trusses using Big Bang-Big Crunch optimization", *J. Struct. Eng.*, ASCE, **133**, 999-1008.

Camp, C.V. and Bichon B.J. (2004), "Design of space trusses using ant colony optimization", *J. Struct. Eng.*, ASCE, **130**, 741-751.

Chapman and Hall/CRC, Computer & Information Science Series, Chapman & Hall/CRC, UK

Coello, C.A.C. (2000), "Use of a self-adaptive penalty approach for engineering optimization problems", *Comput. Indust. Eng.*, **41**, 113-127.

Coello, C.A.C. and Montes, E.M. (2002), "Constraint-handling in genetic algorithms through the use of dominance-based tournament", *IEEE Tran. Reliab.*, **41**, 576-582.

Deb, K. (1991), "Optimal design of a welded beam via genetic algorithms", *AIAA J.*, **29**, 2013-2015.

Deb, K. (2000), "An efficient constraint handling method for genetic algorithms", *Comput. Meth. Appl. Mech. Eng.*, **186**(2-4), 311-338.

Dog, B. and Ö lmez T. (2015), "A new meta-heuristic for numerical function optimization: Vortex Search algorithm", *Inform. Sci.*, **293**, 125-145.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), "The ant system: optimization by a colony of cooperating agents", *IEEE Tran. Syst. Man. Cyber B*, **26**, 29-41.

Eberhart, R.C. and Kennedy J. (1995), "A new optimizer using particle swarm theory", *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan.

Erbatur, F., Hasançebi, O., Tütüncü, I. and Kiliç, H. (2014), "Optimal design of planar and space structures with genetic algorithms", *Comput. Struct.*, **75**, 209-224.

Erol, O.K. and Eksin, I. (2006), "New optimization method: Big Bang-Big Crunch", *Adv. Eng. Softw.*, **37**, 106-111.

Gallagher, R.H., Ragsdell, K.M. and Zienkiewicz, O.C. (1984), *New directions in optimum structural design*, John Wiley, New York.

Gholizadeh, S. and Poorhoseini, H. (2015) "Optimum design of steel frame structures by a modified dolphin echolocation algorithm", *Struct. Eng. Mech.*, **55**, 535-554.

Gholizadeh, S., Gheyratmand, C. and Davoudi, H. (2016), "Optimum design of double layer barrel vaults considering nonlinear behavior", *Struct. Eng. Mech.*, **58**, 1109-1126.

Gonzalez, T.F. (2007), *Handbook of approximation algorithms and meta-heuristics*.

Hasançebi, O., Ç arbas, S., Dogan, E., Erdal, F. and Saka, M.P. (2009), "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures", *Comput. Struct.*, **87**, 284-302.

He, Q. and Wang, L. (2007), "An effective co-evolutionary particle swarm optimization for constrained engineering design problem", *Eng. Appl. Artific. Intell.*, **20**, 89-99.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Hooke, R. and Jeeves, T.A. (1961), "Direct search solution of numerical and statistical problems", *J. Assoc. Comput. Mach.*, **8**(2), 212-229.

Kaveh, A. and Bakhshpoori, T. (2016) "Water evaporation optimization: a novel physically inspired optimization algorithm", *Comput. Struct.*, **167**, 69-85.

Kaveh, A. and Farhoudi, N. (2013), "A new optimization method: dolphin echolocation", *Adv. Eng. Softw.*, **59**, 53-70.
Kaveh, A. and Ilchi Ghazaan, M. (2014), "Enhanced colliding bodies optimization for design problems with continuous and discrete variables", *Adv. Eng. Softw.*, **77**, 66-75.
Kaveh, A. and Khayatazad, M. (2012), "A novel meta-heuristic method: ray optimization", *Comput. Struct.*, **112-113**, 283-294.
Kaveh, A. and Mahdavi, V.R. (2014a), "Colliding bodies optimization: A novel meta-heuristic method", *Comput. Struct.*, **139**, 18-27.
Kaveh, A. and Mahdavi, V.R. (2014b), "Colliding bodies optimization method for optimum design of truss structures with continuous variables", *Adv. Eng. Softw.*, **70**, 1-12.
Kaveh, A. and Mahdavi, V.R. (2014c), "Colliding Bodies Optimization method for optimum discrete design of truss structures", *Comput. Struct.*, **139**, 43-53.
Kaveh, A. and Mahdavi, V.R. (2015), *Colliding Bodies Optimization; Extensions and Applications*, Springer International Publishing, Switzerland.
Kaveh, A. and Talatahari, S. (2010), "A novel heuristic optimization method: charged system search", *Acta Mech.*, **213**, 267-289.
Kaveh, A., Ilchi Ghazaan, M. and Bakhshpoori, T. (2013), "An improved ray optimization algorithm for design of truss structures", *Period Polytech.*, **57**(2), 1-15.
Khot, N.S. and Berke, L. (1984), *Structural optimization using optimality criteria methods*.
Mirjalili, S., Lewis, A. (2016), "The whale optimization algorithm", *Adv. Eng. Softw.*, **95**, 51-67.
Montes, E.M. and Coello, C.A.C. (2008), "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems", *Int. J. General. Syst.*, **37**, 443-473.
Perez, R.E. and Behdinan, K. (2007), "Particle swarm approach for structural design optimization", *Comput. Struct.*, **85**, 1579-1588.
Pianosi, F. and Wagener, T. (2015), "A simple and efficient method for global sensitivity analysis based on cumulative distribution functions", *Environ. Modell. Softw.*, **67**, 1-11.
Plischke, E., Borgonovo, E. and Smith, C.L. (2012), "Global sensitivity measures from given data", *Eur. J. Oper. Res.*, **226**, 536-550.
Ragsdell, K.M. and Phillips, D.T. (1976), "Optimal design of a class of welded structures using geometric programming", *ASME J. Eng. Indust.*, **98**(Ser. B), 1021-1025.
Rahman, S. (2011), "Global sensitivity analysis by polynomial dimensional decomposition", *Reliab. Eng. Syst. Saf.*, **96**, 825-37.
Rastrigin, L.A. (1963), "The convergence of the random search method in the extremal control of a many parameter system", *Autom. Remote Control*, **24**(10), 1337-1342.
Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M. (2012) "Mine blast algorithm for optimization of truss structures with discrete variable", *Comput. Struct.*, **102**, 49-63.
Saka. M.P. (1990), "Optimum design of pin-jointed steel structures with practical applications", *J. Struct. Eng.*, ASCE, **116**, 2599-2620.
Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M. and Tarantola, S. (2010), "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index", *Comput. Phys. Commun.*, **181**, 259-270.
Soh, C.K. and Yang, J. (1996), "Fuzzy controlled genetic algorithm search for shape optimization", *J. Comput. Civil Eng.*, ASCE, **10**, 143-150.
Talbi, E.G. (2009), *Metaheuristics: From Design to Implementation*, Wiley & Sons, Hoboken, New Jersey, USA.
Yang, X.S. (2010), *Nature-inspired meta-heuristic algorithms*, Luniver Press, 2nd Edition, UK.
Zhai, Q., Yang, J. and Zhao, Y. (2014), "Space-partition method for the variance-based sensitivity analysis: Optimal partition scheme and comparative study", *Reliab. Eng. Syst. Saf.*, **131**, 66-82.

CC