IJST, Transactions of Civil Engineering, Vol. 37, No. C1, pp 1-15
Printed in The Islamic Republic of Iran, 2013

OPTIMUM DESIGN OF SPACE TRUSSES USING CUCKOO SEARCH ALGORITHM WITH LÉVY FLIGHTS*

A. KAVEH** AND T. BAKHSHPOORI

Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering,
Iran University of Science and Technology, Narmak, Tehran-16, I. R. of Iran
Email: alikaveh@iust.ac.ir

Abstract– In this paper optimum design of truss structures for both discrete and continuous variables based on the Cuckoo Search (CS) algorithm is presented. The CS is one of the recently developed population based algorithms inspired by the behavior of some cuckoo species together with the Lévy flight behavior of some birds and fruit flies. In order to demonstrate the effectiveness and robustness of the present method, minimum weight design of truss structures is performed and the results of the CS and the selected well-known meta-heuristic search algorithms are compared for both discrete and continuous design of three benchmark truss structures.

Keywords– Optimal design, meta-heuristic search, cuckoo search algorithm, truss structures

1. INTRODUCTION

Optimum design of steel structures with time and resource limitations has always been an important issue for structural designers. Methods employed in structural optimization design problems can be divided into mathematical programming and meta-heuristic algorithms. Due to the difficulties involved in the use of mathematical programming (complex derivatives, sensitivity to initial values, and the large amount of enumeration memory required) [1], for complicated problems, researchers have implemented different kinds of meta-heuristic algorithms for optimum design of structures. These methods consist of : Genetic algorithms (GAs) inspired from Darwin's natural selection theorem, which is based on the idea of the survival of the fittest [2]; Ant Colony Optimization (ACO) that is a cooperative search technique mimicing the foraging behavior of the real-life ant colonies [3]; Particle Swarm Optimizer (PSO) motivated from the social behavior of bird flocking and fish schooling; Harmony Search (HS) algorithm being conceptualized using the musical process of searching for a perfect state of harmony [1, 4], Big Bang-Big Crunch (BB-BC) algorithm that relies on one of the theories of the evolution of the universe [5]; Charged System Search (CSS) method utilizing the governing laws of physics and mechanics [6, 7]; Imperialist Competitive Algorithm (ICA) being a socio-politically motivated optimization algorithm [8]; and hybrid or enhanced meta-heuristic algorithms [9-12].

In this paper, a meta-heuristic method, the so-called Cuckoo Search algorithm, is utilized to determine optimum design of truss structures for both discrete and continuous variables. This algorithm was recently developed by Yang and Deb [13-15], and is based on the obligate brood parasitic behavior of some cuckoo species together with the Lévy flight behavior of some birds and fruit flies. The CS is a population based optimization algorithm and similar to many others meta-heuristic algorithms starts with a random initial population which is taken as host nests or eggs. The CS algorithm essentially works with three components: selection of the best by keeping the best nests or solutions; replacement of the host eggs

*Received by the editors June 28, 2011; Accepted December 13, 2011.
**Corresponding author

with respect to the quality of the new solutions or Cuckoo eggs produced based randomization via Lévy flights globally (exploration); and discovery of some cuckoo eggs by the host birds and replacing according to the quality of the local random walks (exploitation) [14].

Optimum design of the truss structures is known as benchmark in the field of optimal design of structures due to the presence of many design variables, large size of the search space, and many constraints. Therefore this can be considered a suitable means to investigate the efficiency of the new algorithms [8]. Comparison of the CS results with those of selected well-known meta-heuristics demonstrates the efficiency of the present algorithm.

2. OPTIMUM DESIGN OF TRUSS STRUCTURES

The aim of optimizing a structure is to find a set of design variables corresponding to the minimum weight satisfying certain constraints. This can be expressed as [8]:

$$\begin{aligned} &\text{Find} \quad \{X\} = \left[\mathrm{x}_1,\ \mathrm{x}_2,\ \ldots,\ \mathrm{x}_{ng}\right], \quad \mathrm{x}_i \in D_i \\ &\text{To minimize} \quad \mathrm{W}(\{\mathrm{X}\}) = \sum_{\mathrm{i}=1}^{\mathrm{ng}} \mathrm{x}_\mathrm{i} \sum_{j=1}^{nm(i)} \rho_j . L_j \\ &\text{Subject to:} \quad \mathrm{g}_j(\{X\}) \le 0 \quad j = 1,\ 2,\ \ldots,\ n \end{aligned} \tag{1}$$

where $\{X\}$ is the set of design variables; ng is the number of member groups in structure (number of design variables), the grouping of members is performed according to the symmetry in the topology of truss; D_i is the set of available values for the design variable x_i, bounded by an upper and lower limit; $\mathrm{W}(\{\mathrm{X}\})$ presents weight of the structure; $nm(i)$ is the number of members for the *ith* group; ρ_j and L_j denotes the material density and the length of the *jth* member for *ith* group, respectively; $g_j(\{X\})$ denotes design constraints; and n is the number of the constraints.

D_i can be considered either as a continuous set or as a discrete one. In the continuous problems, the design variables can vary continuously in the optimization process.

$$D_i = \left\{\mathrm{x}_i \mid \mathrm{x}_i \in \left[\mathrm{x}_{i,\ \min}, \mathrm{x}_{i,\ \max}\right]\right\} \tag{2}$$

Where $\mathrm{x}_{i,\ \min}$ and $\mathrm{x}_{i,\ \max}$ are minimum and maximum allowable values for the design variables x_i, respectively. If the design variables represent a selection from a set of parts as

$$D_i = \left(d_{i,\ 1},\ d_{i,\ 2}\ ,\ \ldots,\ d_{i,\ \mathrm{nm}(i)}\right) \tag{3}$$

then the problem can be considered as a discrete one.

In order to handle the constraints, a penalty approach is utilized. In this method, the aim of the optimization is redefined by introducing the cost function as:

$$f_{\cos t}(\{X\}) = (1+\varepsilon_1.\ \upsilon)^{\varepsilon_2} \times\ W(\{X\}), \quad \upsilon = \sum_{j=1}^{n} \max\left[0,\ \mathrm{g}_j(\{X\})\right] \tag{4}$$

Where n represents the number of evaluated constraints for each individual design, and υ denotes the sum of the violations of the design. The constants ε_1 and ε_2 are selected considering the exploration and the exploitation rate of the search space. Here, ε_1 is set to unity, ε_2 is selected in a way that it decreases the penalties and reduces the cross-sectional areas. Thus, in the first steps of the search process, ε_2 is set to 1.5 and ultimately increased to 3.

The constraint conditions for truss structures are briefly explained in the following. The stress limitations of the members are imposed according to the provisions of ASD-AISC [16] as follows:

$$\begin{cases} \sigma_i^+ = 0.6\ \mathrm{F}_y & \text{for } \sigma_i \geq 0 \\ \sigma_i^- & \text{for } \sigma_i < 0 \end{cases} \tag{5}$$

$$\sigma_i^- = \begin{cases} \left[\left(1 - \dfrac{\lambda_i^2}{2\ \mathrm{c}_c^2}\right) F_y\right] \Big/ \left(\dfrac{5}{3} + \dfrac{3\lambda_i}{8\ c_c} + \dfrac{\lambda_i^3}{8\ \mathrm{c}_c^3}\right) & for\ \lambda_i \geq c_c \\ \dfrac{12\ \pi^2\ E}{23\ \lambda_i^2} & for\ \lambda_i \geq c_c \end{cases} \tag{6}$$

Where, E is the modulus of elasticity; F_y is the yield stress of steel; c_c denotes the slenderness ratio (λ_i) dividing the elastic and inelastic buckling regions ($c_c = \sqrt{2\pi^2 E / F_y}$); $\lambda_i =$ the slenderness ratio$\left(\lambda_i = kl_i / r_i\right)$; $k =$ the effective length factor; $L_i =$ the member length; and $r_i =$ the radius of gyration. The radius of gyration (r_i) can be expressed in terms of cross-sectional areas as $r_i = a\ A_i^b$. Here, *a* and *b* are the constants depending on the types of sections adopted for the members such as pipes, angles, and tees. In this study, pipe sections $(a = 0.4993 \text{ and } b = 0.6777)$ are adopted for bars [1].

The other constraint corresponds to the limitation of the nodal displacements:

$$\delta_i - \delta_i^u \leq 0 \qquad i = 1,\ 2,\ \ldots,\ nn \tag{7}$$

Where δ_i is the nodal deflection; δ_i^u is the allowable deflection of node *i*; and nn is the number of nodes.

3. LÉVY FLIGHTS AS RANDOM WALKS

The randomization plays an important role in both exploration and exploitation in meta-heuristic algorithms. The Lévy flights as random walks can be described as follows [13]:

A random walk is a random process which consists of taking a series of consecutive random steps. A random walk can be expressed as:

$$S_n = \sum_{i=1}^{n} X_i = X_1 + X_2 + \ \ldots + X_n = \sum_{i=1}^{n-1} X_i + X_n = S_{n-1} + X_n \tag{8}$$

Where S_n presents the random walk with *n* random steps and X_i is the *i*th random step with predefined length. The last statement means that the next state will only depend on the current existing state and the motion or transition X_n. In fact, the step size or length can vary according to a known distribution. A very special case is when the step length obeys the Lévy distribution; such a random walk is called a Lévy flight or Lévy walk. Actually, Lévy flights have been observed among foraging patterns of albatrosses, fruit flies and spider monkeys.

From the implementation point of view, the generation of random numbers with Lévy flights consists of two steps: the choice of a random direction and the generation of steps which obey the chosen Lévy distribution. While the generation of steps is quite tricky, there are a few ways of achieving this. One of the most efficient and yet straightforward ways is to use the so-called Mantegna algorithm. In the Mantegna's algorithm, the step length *S* can be calculated by:

$$S = \frac{u}{|v|^{1/\beta}} \tag{9}$$

Where β is a parameter between [1, 2] interval and is considered to be 1.5; *u* and *v* are drawn from normal distribution as

$$u \sim N(0,\ \sigma_u^2), \qquad v \sim N(0, \sigma_v^2) \tag{10}$$

Where

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\ \sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\ \beta\ 2^{(\beta-1)/2}} \right\}^{1/\beta}, \qquad \sigma_v = 1 \tag{11}$$

Studies show that the Lévy fights can maximize the efficiency of the resource searches in uncertain environments.

4. CUCKOO SEARCH ALGORITHM

This algorithm is inspired by some species of a bird family called cuckoo because of their special lifestyle and aggressive reproduction strategy. These species lay their eggs in the nests of other host birds (almost other species) with amazing abilities such as selecting the recently spawned nests, and removing the existing eggs that increase the hatching probability of their eggs. On the other hand, some of the host birds are able to combat this parasites behavior of cuckoos, and throw out the discovered alien eggs or build their new nests in new locations.

This algorithm contains a population of nests or eggs. For simplicity, the following representations are used; where each egg in a nest represents a solution and a Cuckoo egg represents a new one. If the Cuckoo egg is very similar to the host's egg, then this Cuckoo's egg is less likely to be discovered, thus the fitness should be related to the difference in solutions. The aim is to employ new and potentially better solutions (Cuckoos') to replace a not-so-good solution in the nests [14].

For simplicity in describing the CS, the following three idealized rules are utilized [15]:

1) Each Cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;

2) The best nests with high quality of eggs are carried over to the next generations;

3) The number of available host nests is constant, and the egg which is laid by a Cuckoo is discovered by the host bird with a probability of *pa* in the range of [0, 1]. The later assumption can be approximated by the fraction *pa* of the *n* nests which is replaced by new ones (with new random solutions).

Based on the above three rules, the basic steps of the CS can be summarized as the pseudo code shown in Fig. 1.

This pseudo code, provided in the book entitled Nature-Inspired meta-heuristic algorithms by Yang [13], is a sequential version and each iteration of the algorithm consists of two main steps, but another version of the CS which is supposed to be different and more efficient is provided by Yang and Deb [15]. This new version has some differences with the book version as follows:

Objective function $f(x),\ x=(x_1,\ x_2,\ \dots,\ x_d)$;
Generate initial population of n host nests $x_i\ (i=1,\ 2,\ \dots,\ n)$;
while (stop criterion)
  Get a Cuckoo randomly by Lévy flights;
  Evaluate its quality/fitness F_i;
  Choose a nest among n (say j) randomly;
  if $F_i \geq F_j$
    replace j by the new solution;
  end
  Abandon a fraction (*pa*) of worse nests
  [and build new ones at new locations via Lévy flights]
  Keep the best solutions (or nests with quality solutions);
  Rank the solutions and find the current best;
end while
Post process results and visualization;

Fig. 1. Pseudo code of the CS

In the first step, according to the pseudo code, one of the randomly selected nests (except the best one) is replaced by a new solution produced by random walk with Lévy flight around the so far best nest, considering the quality. But in the new version, all of the nests except the best one are replaced in one step, by new solutions. When generating new solutions, $x_i^{(t+1)}$ for the *i*th Cuckoo, a Lévy flight is performed using the following equation:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha . S \tag{12}$$

where $\alpha > 0$ is the step size parameter and should be chosen considering the scale of the problem and is set to unity in the CS [14] and decreases function as the number of generations increases in the modified CS [17, 18]. It should be noted that in this new version, the solutions' current positions are used instead of the best solution so far as the origin of the Lévy flight. The step size is considered as 0.1 in this work because it results in efficient performance of algorithm in our examples. The parameter *S* is the length of random walk with Lévy flights according to the Mantegna's algorithm as described in the Eq. (9).

In the second step, the *pa* fraction of the worst nests are discovered and replaced by new ones. However, in the new version, the parameter *pa* is considered as the probability of a solution's component to be discovered. Therefore, a probability matrix is produced as:

$$P_{ij} = \begin{cases} 1 & if\ \ rand < pa \\ 0 & if\ \ rand \geq pa \end{cases} \tag{13}$$

Where *rand* is a random number in [0, 1] interval and P_{ij} is discovering probability for *j*th variable of *i*th nest. Then all of the nests are replaced by new ones produced by random walks (point wise multiplication of random step sizes with probability matrix) from their current positions according to quality. In this study the later version of the CS algorithm is used for optimum design of truss structures.

5. OPTIMUM DESIGN OF TRUSS STRUCTURES USING CUCKOO SEARCH ALGORITHM

The pseudo code of optimum design algorithm is as follows:

a) Initialize the Cuckoo Search algorithm parameters

The CS parameters are set in the first step. These parameters are number of nests (n), step size parameter (α), discovering probability (pa) and maximum number of analyses as the stopping criterion.

b) Generate initial nests or eggs of host birds

The initial locations of the nests are determined by the set of values assigned to each decision variable randomly as

$$nest_{i,j}^{(0)} = x_{j,\min} + rand.(x_{j,\max} - x_{j,\min}) \quad (14)$$

Where $nest_{i,j}^{(0)}$ determines the initial value of the *j*th variable for the *i*th nest; $x_{j,min}$ and $x_{j,max}$ are the minimum and the maximum allowable values for the *j*th variable; *rand* is a random number in the interval [0, 1]. For problems with discrete design variables it is necessary to use a rounding function.

c) Generate new Cuckoos by Lévy flights

In this step all of the nests except for the best so far are replaced in order of quality by new Cuckoo eggs produced with Lévy flights from their positions as

$$nest_i^{(t+1)} = nest_i^{(t)} + \alpha \,.\, S.(nest_i^{(t)} - nest_{best}^{(t)}).r \quad (15)$$

where $nest_i^t$ is the *ith* nest current position; α is the step size parameter which is considered to be 0.1; *S* is the Lévy flights vector as in Mantegna's algorithm; *r* is a random number from a standard normal distribution and $nest^t_{best}$ is the position of the best nest so far.

d) Alien eggs discovery

The alien eggs discovery is performed for all of the eggs using the probability matrix for each component of each solution. Existing eggs are replaced considering quality by newly generated ones from their current position by random walks with step size such as [19]:

$$\begin{aligned} &S = rand\,.\big(nests\left[permute1[i][j]\right] - nests\left[permute2[i][j]\right]\big) \\ &nest^{(t+1)} = nest^{(t)} + S.*P \end{aligned} \quad (16)$$

where *permute1* and *permute2* are different rows permutation functions applied to the nests matrix and *P* is the probability matrix which was mentioned in the Eq. (13).

e) Termination criterion

The generation of new Cuckoos and the discovery of the alien eggs steps are performed alternately until a termination criterion is satisfied. The maximum number of structure analyses is considered as the algorithm's termination criterion.

6. DESIGN EXAMPLES

In this section, common truss optimization examples as benchmark problems are optimized with the CS algorithm. The final results are compared to the solutions of other methods to demonstrate the efficiency of the CS. We have tried to vary the number of host nests (or the population size of n) and the probability pa. From our simulations, we found that *n=7 to 20* and *pa=0.15 to 0.35* are efficient for design examples. The examples contain a 25-bar transmission tower and a 72-bar spatial truss with both discrete and continuous design variables and a dome shaped space truss with continuous search space.

Example 1: A 25-bar space truss

The 25-bar transmission tower is widely used in structural optimization to verify various meta-heuristic algorithms. The topology and nodal numbering of a 25-bar space truss structure is shown in Fig. 2 taken from [6]. The material density is considered as 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is taken as 10^7 psi (68,950 MPa). Twenty-five members are categorized into eight groups, as follows: (1) A_1, (2) A_2 –A_5, (3) A_6–A_9, (4) A_{10}–A_{11}, (5) A_{12}–A_{13}, (6) A_{14}–A_{17}, (7) A_{18}–A_{21}, and (8) A_{22}–A_{25}. In this example, designs for both a single and multiple load cases using both discrete and continuous design variables are performed. The parameters of the CS algorithm are considered to be pa=0.15, number of nests=10 and the maximum number of analyses=14,000 as the stopping criterion.

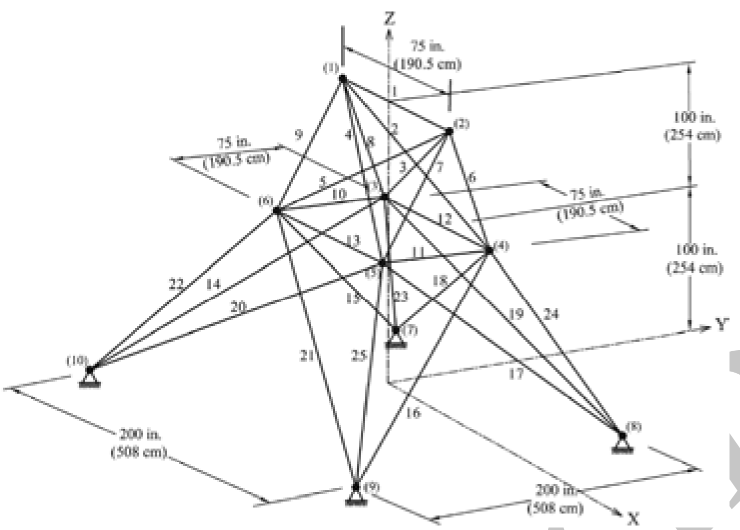

Fig. 2. A 25-bar space truss

i. Design of the 25-bar truss utilizing discrete variables

In the first design of the 25-bar truss, a single load case {(kips) (kN)} is applied to the structure at nodes 1, 2, 3 and 4 as follows: 1{(0, −10, −10) (0, −44.5, −44.5)}, 2{(1, −10, −10) (4.45, −44.5, −44.5)}, 3{(0.6, 0, 0) (2.67, 0, 0)} and 4{(0.5, 0, 0) (2.225, 0, 0)}. The allowable stresses and displacements are respectively ±40 ksi (275.80 MPa) for each member and ±0.35 in (±8.89 mm) for each node in the x, y and z directions. The range of discrete cross-sectional areas is from 0.1 to 3.4 in^2 (0.6452 to 21.94 cm^2) with 0.1 in^2 (0.6452 cm^2) increment (resulting in 34 discrete cross sections) for each of the eight element groups [20].

The CS algorithm achieves the best solution weighted by 484.85 lb (2157.58 N), after 2000 analyses. Although this is identical to the best design developed using BB-BC algorithm [20] and a multiphase ACO procedure [21], it performs better than others when the number of analyses and average weight for 100 runs are compared. Table 1 presents the performance of the CS and other heuristic algorithms.

Table 1 Performance comparison for the 25-bar spatial truss under single load case

Element group		Optimal cross-sectional areas (in^2)					
		GA	GA	ACO	BB-BC phase 1, 2	Present work	
		[20]	[20]	[21]	[20]	in^2	cm^2
1	A_1	0.10	0.10	0.10	0.10	0.10	0.645
2	A_2–A_5	1.80	0.50	0.30	0.30	0.30	1.935
3	A_6–A_9	2.30	3.40	3.40	3.40	3.40	21.935
4	A_{10}–A_{11}	0.20	0.10	0.10	0.10	0.10	0.645
5	A_{12}–A_{13}	0.10	1.90	2.10	2.10	2.10	13.548
6	A_{14}–A_{17}	0.80	0.90	1.00	1.00	1.00	6.452
7	A_{18}–A_{21}	1.80	0.50	0.50	0.50	0.50	3.226
8	A_{22}–A_{25}	3.00	3.40	3.40	3.40	3.40	21.935
Best weight (lb)		546.01	485.05	484.85	484.85	484.85	2157.58 (N)
Average weight (lb)		N/A	N/A	486.46	485.10	485.01	2158.29 (N)
Number of analyses		800	15,000	7700	9000	2000	

ii. Design of the 25-bar truss utilizing continuous variables

In the second design of the 25-bar truss, the structure is subjected to two load cases listed in Table 2. Maximum displacement limitations of ± 0.35 in (±8.89 mm) are imposed on every node in every direction and the axial stress constraints vary for each group as shown in Table 3. The range of cross-sectional areas varies from 0.01 to 3.4 in^2 (0.06452 to 21.94 cm^2) [6].

Table 4 shows the best solution vectors, the corresponding weights, average weights and the required number of analyses for the present algorithm and some other meta-heuristic algorithms. The best result is obtained by IACS algorithm [22] from the point of low weight and number of analyses. The CS-based algorithm needs 6100 analyses to find the best solution while this number is equal to 9596, 15,000, 9875, 12,500 and 7000 analyses for a PSO-based algorithm [6], HS algorithm [1], a combination algorithm based on PSO, ACO and HS [10], an improved BB–BC method using PSO properties [5] and the CSS algorithm [6], respectively. The difference between the result of the CS and these algorithms is very small, but the average weight obtained by the CS algorithm for 100 runs is better than others. The convergence history for best result and average weight of 100 runs is shown in Fig. 3. The important point is that although the CS requires 6100 analyses to achieve the 545.17 lb (2426.02 N), it can achieve the 545.76 lb (2428.63 N) after 2700 analyses, because CS uses the exploration step in terms of Lévy flights. If the search space is large, Lévy flights are usually more efficient.

Table 2. Loading conditions for the 25-bar spatial truss

Case	Node	F_x kips (kN)	F_y kips (kN)	Fz kips (kN)
1	1	1.0 (4.45)	10.0 (44.5)	−5.0 (−22.25)
	2	0.0	10.0	−5.0 (−22.25)
	3	0.5 (2.225)	0.0	0.0
	6	0.5 (2.225)	0.0	0.0
2	1	0.0	20.0 (89)	−5.0 (−22.25)
	2	0.0	−20.0 (−89)	−5.0 (−22.25)

Table 3. Member stress limitation for the 25-bar space truss

Element group		Compression ksi (MPa)	Tension ksi (MPa)
1	A_1	35.092 (241.96)	40.0 (275.80)
2	A_2–A_5	11.590 (79.913)	40.0 (275.80)
3	A_6–A_9	17.305 (119.31)	40.0 (275.80)
4	A_{10}–A_{11}	35.092 (241.96)	40.0 (275.80)
5	A_{12}–A_{13}	35.092 (241.96)	40.0 (275.80)
6	A_{14}–A_{17}	6.759 (46.603)	40.0 (275.80)
7	A_{18}–A_{21}	6.959 (47.982)	40.0 (275.80)
8	A_{22}–A_{25}	11.082 (76.410)	40.0 (275.80)

Table 4. Performance comparison for the 25-bar spatial truss under multiple load cases

Element group		Optimal cross-sectional areas (in^2)							
		PSO [6]	HS [1]	IACS [22]	HPSACO [10]	HBB–BC [5]	CSS [6]	Present work	
								in^2	cm^2
1	A_1	0.010	0.047	0.010	0.010	0.010	0.010	0.01	0.065
2	A_2–A_5	2.121	2.022	2.042	2.054	1.993	2.003	1.979	12.765
3	A_6–A_9	2.893	2.950	3.001	3.008	3.056	3.007	3.005	19.386
4	A_{10}–A_{11}	0.010	0.010	0.010	0.010	0.010	0.010	0.01	0.065
5	A_{12}–A_{13}	0.010	0.014	0.010	0.010	0.010	0.010	0.01	0.065
6	A_{14}–A_{17}	0.671	0.688	0.684	0.679	0.665	0.687	0.686	4.428
7	A_{18}–A_{21}	1.611	1.657	1.625	1.611	1.642	1.655	1.679	10.830
6	A_{22}–A_{25}	2.717	2.663	2.672	2.678	2.679	2.660	2.656	17.134
Best weight (lb)		545.21	544.38	545.03	544.99	545.16	545.10	545.17	2426.02 (N)
Average weight (lb)		546.84	N/A	545.74	545.52	545.66	545.58	545.18	2426.05 (N)
Number of analyses		9596	15,000	3254	9875	12,500	7000	6100	

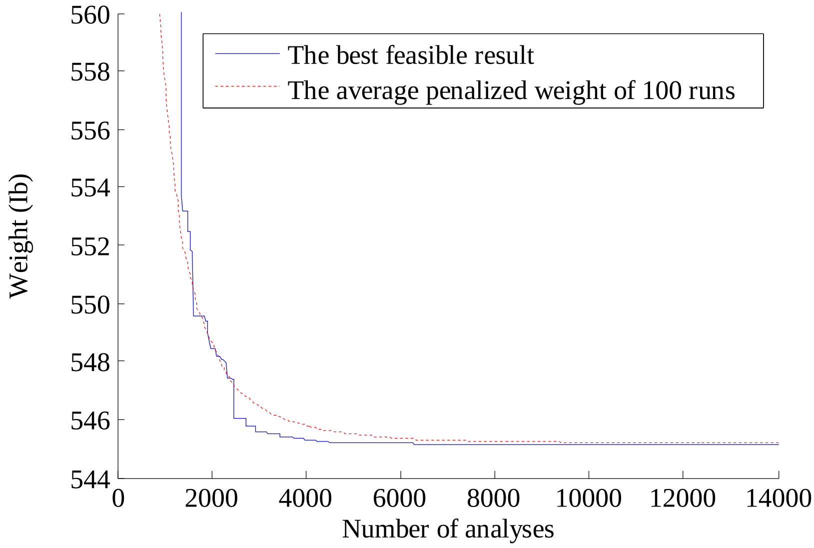

Fig. 3. Convergence history of the 25-bar space truss under multiple load cases

Example 2: A 72- bar space truss

For the 72-bar spatial truss structure shown in Fig. 4 taken from [5], the material density is 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is 10^7 psi (68,950 MPa). The 72 structural members of this spatial truss are categorized into 16 groups using symmetry: (1) A_1–A_4, (2) A_5–A_{12}, (3) A_{13}–A_{16}, (4) A_{17}–A_{18}, (5) A_{19}– A_{22}, (6) A_{23}–A_{30}, (7) A_{31}–A_{34}, (8) A_{35}–A_{36}, (9) A_{37}–A_{40}, (10) A_{41}–A_{48}, (11) A_{49}–A_{52}, (12) A_{53}–A_{54}, (13) A_{55}–A_{58}, (14) A_{59}–A_{66} (15), A_{67}– A_{70}, and (16) A_{71}–A_{72}. In this example, designs for multiple load cases using both discrete and continuous design variables are performed. The values and directions of the two load cases applied to the 72-bar spatial truss for both discrete and continuous designs are listed in Table 5. The members are subjected to the stress limits of ±25 ksi (±172.375 MPa) for both discrete and continuous designs. Maximum displacement limitations of ±0.25 in (±6.35 mm) are imposed on every node in every direction and on the uppermost nodes in both x and y directions respectively for discrete and continuous cases. In this example, the parameters of the CS algorithm are considered to be pa=0.15 and number of nests =7, maximum number of analyses=21,000.

Table 5. Multiple loading conditions for the 72-bar truss

Case	Node	F_x kips (kN)	F_y kips (kN)	F_z kips (kN)
1	17	0.0	0.0	−5.0 (−22.25)
	18	0.0	0.0	−5.0 (−22.25)
	19	0.0	0.0	−5.0 (−22.25)
	2	0.	0.0	−5.0 (−22.25)
2	17	5.0 (22.25)	5.0 (22.25)	−5.0 (−22.25)

i. Design of the 72-bar truss using discrete variables

In this case, the discrete variables are selected from 64 discrete values from 0.111 to 33.5 in^2 (71.613 to 21612.860 mm^2). For more information, the reader can refer to Table 2 in Kaveh and Talatahari [8].

Table 6 shows the best solution vectors, the corresponding weights and the required number of analyses for the present algorithm and some other meta-heuristic algorithms. The CS algorithm can find the best design among the other existing studies. Although the number of required analyses by the CS algorithm is slightly more than ICA algorithm, the best weight of the CS algorithm is 389.87 lb (1734.93 N), that is 2.97 lb (13.22 N), lighter than the best result obtained by ICA algorithm [8].

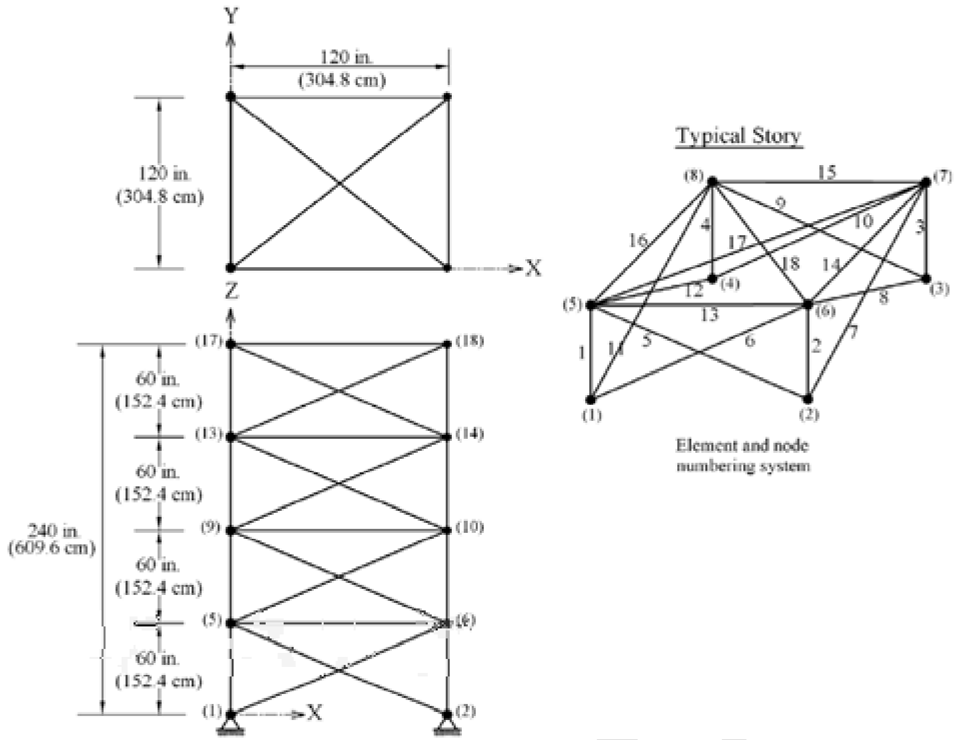

Fig. 4. A 72-bar space truss

Table 6. Performance comparison for the 72-bar spatial truss with discrete variables

Element group		Optimal cross-sectional areas (in^2)						
		GA [8]	PSOPC [8]	HPSO [8]	HPSACO [9]	ICA [8]	Present work in^2	cm^2
1	A_1–A_4	0.196	4.490	4.970	1.800	1.99	1.800	11.613
2	A_5–A_{12}	0.602	1.457	1.228	0.442	0.442	0.563	3.632
3	A_{13}–A_{16}	0.307	0.111	0.111	0.141	0.111	0.111	0.716
4	A_{17}–A_{18}	0.766	0.111	0.111	0.111	0.141	0.111	0.716
5	A_{19}–A_{22}	0.391	2.620	2.880	1.228	1.228	1.266	8.168
6	A_{23}–A_{30}	0.391	1.130	1.457	0.563	0.602	0.563	3.632
7	A_{31}–A_{34}	0.141	0.196	0.141	0.111	0.111	0.111	0.716
8	A_{35}–A_{36}	0.111	0.111	0.111	0.111	0.141	0.111	0.716
9	A_{37}–A_{40}	1.800	1.266	1.563	0.563	0.563	0.563	3.632
10	A_{41}–A_{48}	0.602	1.457	1.228	0.563	0.563	0.442	2.852
11	A_{49}–A_{52}	0.141	0.111	0.111	0.111	0.111	0.111	0.716
12	A_{53}–A_{54}	0.307	0.111	0.196	0.250	0.111	0.111	0.716
13	A_{55}–A_{58}	1.563	0.442	0.391	0.196	0.196	0.196	1.265
14	A_{59}–A_{66}	0.766	1.457	1.457	0.563	0.563	0.602	3.884
15	A_{67}–A_{70}	0.141	1.228	0.766	0.442	0.307	0.391	2.523
16	A_{71}–A_{72}	0.111	1.457	1.563	0.563	0.602	0.563	3.632
Weight (lb)		427.203	941.82	933.09	393.380	392.84	389.87	1734.93 (N)
Number of analyses		N/A	150,000	50,000	5330	4500	4840	

ii. Design of the 72-bar truss using continuous variables

In this case the minimum value for the cross-Sectional areas is 0.1 in^2 (0.6452 cm^2) and the maximum value is limited to 4.00 in^2 (25.81 cm^2).

The CS algorithm achieves the best result among other algorithms from the aspects of weight, number of required analyses and the average weight of 100 runs. The convergence history of the best result and the average weight of 100 runs are shown in Fig. 5. Notice that as shown in this figure, although the CS requires 10,600 analyses to achieve 379.63 lb (1689.37 N), it achieves the 380 lb (1691 N) possible design after 4900 analyses. Table 7 compares the results of the CS to those of the previously reported methods in the literature.

For further studies of one of two CS parameters we have tried this example alternatively for constant number of nests as 7 and various amounts of *pa* from the [0, 1] interval with 21,000 as the maximum number of analyses. The convergence history of the average weight for 100 runs is shown in Fig. 6. According to this figure, the values from [0.15, 0.35] are more efficient for the performance of the algorithm and 0.15 gives the best result among others.

Table 7. Performance comparison for the 72-bar spatial truss with continuous variables

Element group		Optimal cross-sectional areas (in^2)						
		GA [5]	ACO [21]	PSO [5]	BB–BC [20]	HBB–BC [5]	Present work in^2	cm^2
1	A_1–A_4	1.755	1.948	1.7427	1.8577	1.9042	1.9122	12.055
2	A_5–A_{12}	0.505	0.508	0.5185	0.5059	0.5162	0.5101	3.267
3	A_{13}–A_{16}	0.105	0.101	0.1000	0.1000	0.1000	0.1000	0.646
4	A_{17}–A_{18}	0.155	0.102	0.1000	0.1000	0.1000	0.1000	0.645
5	A_{19}–A_{22}	1.155	1.303	1.3079	1.2476	1.2582	1.2577	8.487
6	A_{23}–A_{30}	0.585	0.511	0.5193	0.5269	0.5035	0.5128	3.343
7	A_{31}–A_{34}	0.100	0.101	0.1000	0.1000	0.1000	0.1000	0.645
8	A_{35}–A_{36}	0.100	0.100	0.1000	0.1012	0.1000	0.1000	0.646
9	A_{37}–A_{40}	0.460	0.561	0.5142	0.5209	0.5178	0.5229	3.197
10	A_{41}–A_{48}	0.530	0.492	0.5464	0.5172	0.5214	0.5177	3.345
11	A_{49}–A_{52}	0.120	0.100	0.1000	0.1004	0.1000	0.1000	0.648
12	A_{53}–A_{54}	0.165	0.107	0.1095	0.1005	0.1007	0.1000	0.645
13	A_{55}–A_{58}	0.155	0.156	0.1615	0.1565	0.1566	0.1566	1.013
14	A_{59}–A_{66}	0.535	0.550	0.5092	0.5507	0.5421	0.5406	3.492
15	A_{67}–A_{70}	0.480	0.390	0.4967	0.3922	0.4132	0.4152	2.839
16	A_{71}–A_{72}	0.520	0.592	0.5619	0.5922	0.5756	0.5701	3.486
Weight (lb)		385.76	380.24	381.91	379.85	379.66	379.63	1689.37 (N)
Average weight (lb)		N/A	383.16	N/A	382.08	381.85	379.73	1689.80 (N)
Number of analyses		N/A	18,500	N/A	19,621	13,200	10,600	

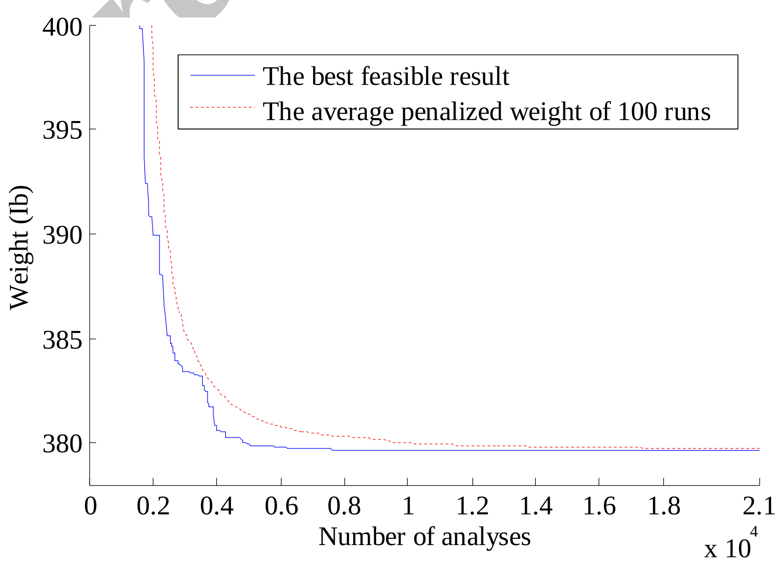

Fig. 5. Convergence history of the 72-bar space truss with continious variables

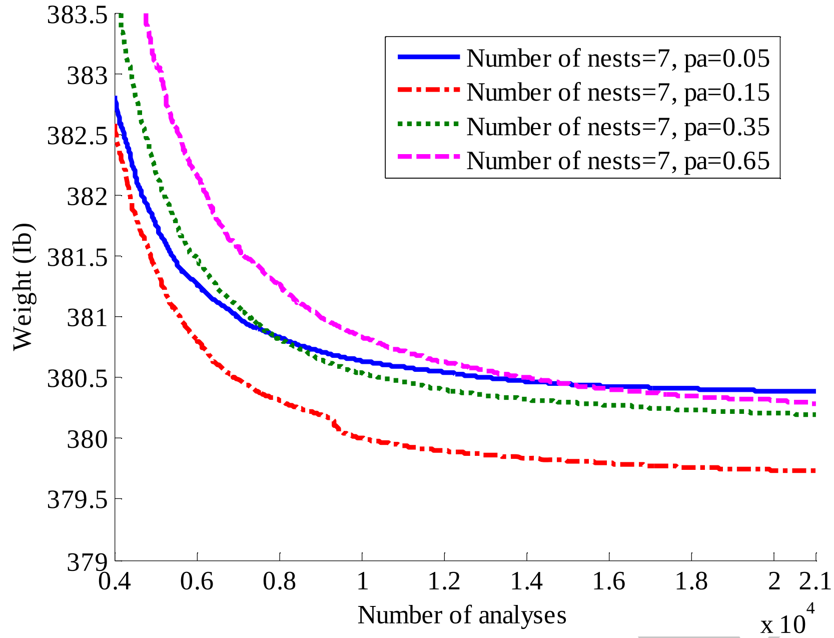

Fig. 6. The convergence history for the average weight of 100 runs, with constant number of nests and different values of *pa*

Example 3: Design of the 120-bar dome shaped truss

The topology, nodal numbering and element grouping of the 120-bar dome truss are shown in Fig. 7. For clarity, not all the element groups are numbered in this figure. The 120 members are categorized into seven groups, because of symmetry. Other conditions of the problem are as follows [8], the modulus of elasticity is 30,450 ksi (210,000 MPa) and the material density is 0.288 lb/in^3 (7971.810 kg/m^3). The yield stress of steel is taken as 58.0 ksi (400 MPa). The dome is considered to be subjected to vertical loading at all the unsupported joints. These loads 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 rest of the nodes. The minimum cross-sectional area of all members is 0.775 in^2 (5 cm^2) and the maximum cross-sectional area is taken as 20.0 in^2 (129.032 cm^2). The constraints are stress constraints (as defined by Eqs. (5) and (6)) and displacement limitations of ±0.1969 in (±5 mm), imposed on all nodes in x, y and z directions.

In this example, the parameters of the CS algorithm are considered to be *pa*=0.15, the number of nests =7 and the maximum number of analyses=21,000. Table 8 shows the best solution vectors, the corresponding weights and the required number of analyses for convergence of the present algorithm and some other meta-heuristic algorithms. The CS-based algorithm needs 6300 analyses to find the best solution while this number is equal to 150,000, 32,600, 10,000, 10,000, 7000 and 6000 analyses for a PSO-based algorithm [10], a PSO and ACO hybrid algorithm [10], a combination algorithm based on PSO, ACO and HS [10], an improved BB–BC method using PSO properties [5], the CSS algorithm [6] and the ICA algorithm [8], respectively. As a result, the CS optimization algorithm has the second best convergence rates among the considered meta-heuristics and its difference with the ICA is only 300 analyses. Comparing the final results of the CS and those of the other meta-heuristics shows that CS finds the second best result while the difference between the result of the CS and that obtained by the HPSACO [10], as the first best result, is very small. A comparison of the allowable and existing stresses and displacements of the 120-bar dome truss structure using CS is shown in Fig. 8. The maximum value for displacement is equal to 0.1969 in (5 mm) and the maximum stress ratio is equal to 99.99%.

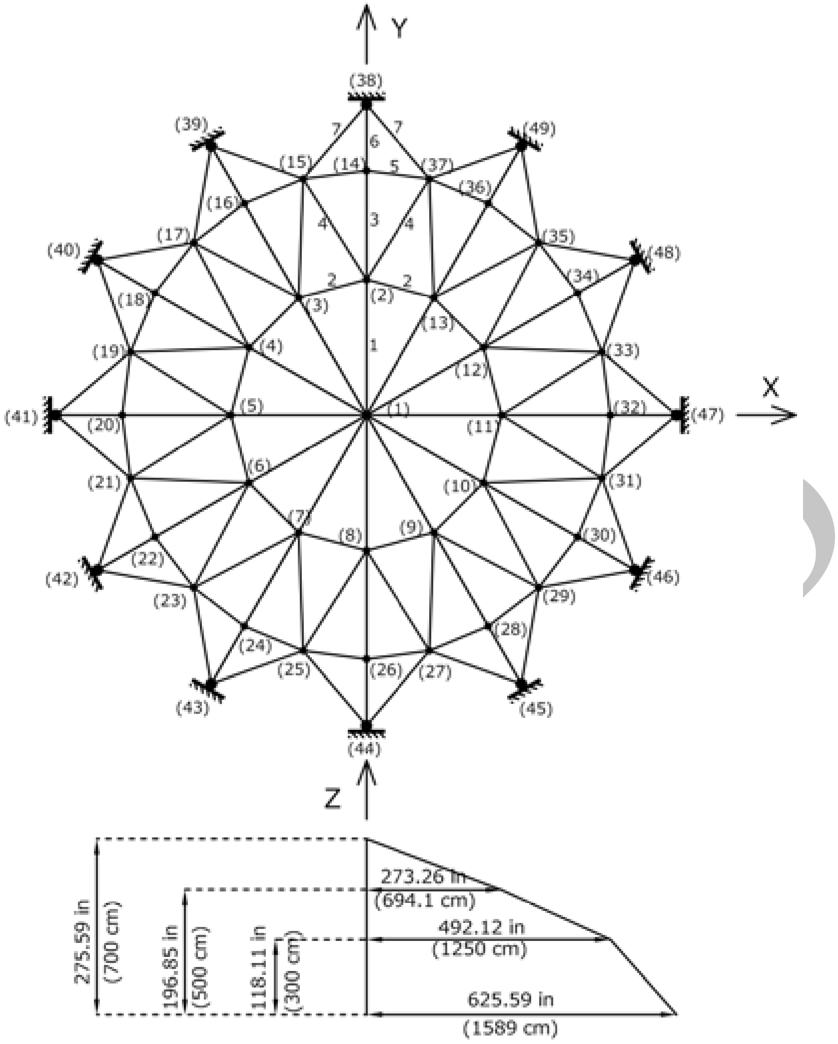

Fig. 7. A 120-bar dome shaped truss

Table 8. Performance comparison for the 120-bar dome shaped truss with continuous variables

Element group		Optimal cross-sectional areas (in^2)							
		PSOPC [10]	PSACO [10]	HPSACO [10]	HBB–BC [5]	CSS [6]	ICA [8]	Present work in^2	Present work cm^2
1	A_1	3.040	3.026	3.095	3.037	3.027	3.0275	3.0244	19.512
2	A_2	13.149	15.222	14.405	14.431	14.606	14.4596	14.7168	94.947
3	A_3	5.646	4.904	5.020	5.130	5.044	5.2446	5.0800	32.774
4	A_4	3.143	3.123	3.352	3.134	3.139	3.1413	3.1374	20.241
5	A_5	8.759	8.341	8.631	8.591	8.543	8.4541	8.5012	54.847
6	A_6	3.758	3.418	3.432	3.377	3.367	3.3567	3.3019	21.303
7	A_7	2.502	2.498	2.499	2.500	2.497	2.4947	2.4965	16.106
Best weight (lb)		33481.2	33263.9	33248.9	33287.9	33251.9	33256.2	33250.42	147964.37 (N)
Average weight(lb)		N/A	N/A	N/A	N/A	N/A	N/A	33253.28	147977.10 (N)
Number of analyses		150,000	32,600	10,000	10,000	7000	6000	6300	

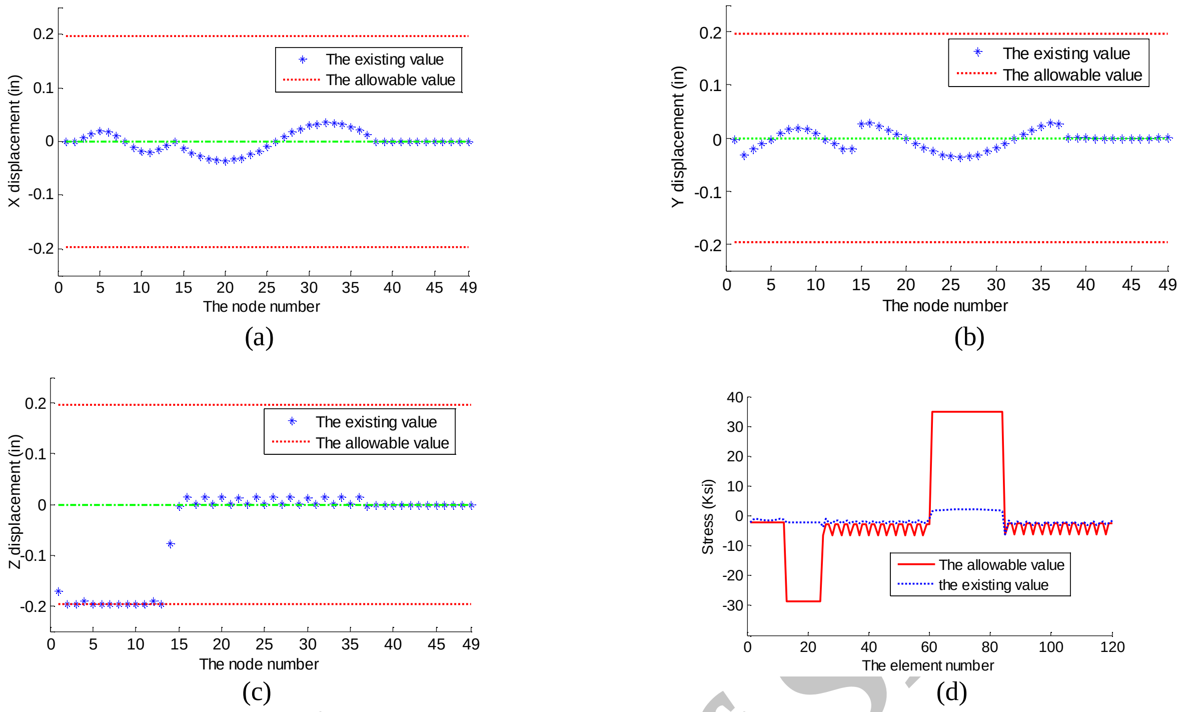

Fig. 8. Comparison of the allowable and existing constraints for the 120-bar dome shaped truss using the CS (a) Displacement in the x direction, (b) Displacement in the y direction, (c) Displacement in the z direction, (d) Stresses

7. CONCLUSION

A version of cuckoo search algorithm via Lévy flights is applied to optimum design of truss structures using both discrete and continuous design variables. Looking at the CS algorithm carefully, one can observe essentially three components: selection of the best, exploitation by local random walk, and exploration by randomization via Lévy flights globally. In order to sample the search space effectively so that the newly generated solutions are diverse enough, the CS uses the exploration step in terms of Lévy flights. In contrast, most meta-heuristic algorithms use either uniform distributions or Gaussian to generate new explorative moves. For large search spaces the Lévy flights are usually more efficient.

Unique characteristics of the CS algorithm over other meta-heuristic methods are its simplified numerical structure and its dependency on a relatively small number of parameters to define and determine - or limit- the algorithm's performance. In fact, apart from the step size parameter α and the population size n , there is essentially one parameter *pa*.

Three design examples consisting of two space trusses with continuous and discrete design variables and a dome-shaped truss with continuous search space are studied to illustrate the efficiency of the present algorithm. The comparisons of the numerical results of these structures utilizing the CS and those obtained by other optimization methods are carried out to demonstrate the robustness of the present algorithm in terms of good results and number of analyses together. The most noticeable result obtained by the CS is that the average weight of 100 runs is better than other algorithms.

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

REFERENCES

1. Lee, K. S. & Geem, W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & Structures*, Vol. 82, pp. 781-798.

2. Kaveh, A. & Malakouti Rad, S. (2010). Hybrid genetic algorithm and particle swarm optimization for the force method-based simultaneous analysis and design. *Iranian Journal of Science and Technology,* Vol. **34**, pp. 15-34.
3. Kaveh, A. & Masoudi, M. S. (2011). Cost optimization of a composite floor system using ant colony system. *Iranian Journal of Science & Technology, Transaction B: Engineering*, In press.
4. Kaveh, A. & Shakouri Mahmud Abadi, S. (2010). Harmony search algorithm for optimum design of slab formwork. *Iranian Journal of Science & Technology, Transaction B: Engineering*, Vol. 34, No. B4, pp. 335-351.
5. Kaveh, A. & Talatahari, S. (2009). Size optimization of space trusses using Big Bang-Big Crunch algorithm. *Computers & Structures*, Vol. 87, pp. 1129-1140.
6. Kaveh, A. & Talatahari, S. (2010). Optimal design of skeletal structures via the charged system search algorithm. *Structural and Multidisciplinary Optimization*, Vol. 41, pp. 893-911.
7. Kaveh, A., Talatahari, S & Farahmand Azar, B. (2011). Optimum design of composite open channels using charged system search algorithm. *Iranian Journal of Science & Technology, Transaction B: Engineering*, Article in press.
8. Kaveh, A. & Talatahari, S. (2010). Optimum design of skeletal structures using imperialist competitive algorithm, *Computers & Structures*, vol. 88, pp. 1220-1229.
9. Kaveh, A. & Talatahari, S. (2009). A particle swarm ant colony optimization for truss structures with discrete variables, *Journal of Constructional Steel Research*, Vol. 65, pp. 1558-1568.
10. Kaveh, A. & Talatahari, S. (2009). Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers & Structures*, Vol. 87, pp. 267-283.
11. Kaveh, A. & Laknejadi, K. (2011). A hybrid multi-objective particle swarm optimization and decision making procedure for optimal design of truss structures, *Iranian Journal of Science and Technology*, Vol. 35, No. C2, pp. 137-154.
12. Kaveh, A. & Talatahari, S. (2010). An improved ant colony optimization for the design of planar steel frames, *Engineering Structures*, Vol. 32, pp. 864-873.
13. Yang, X. S. & Deb, S. (2008). *Nature-Inspired Metaheuristic Algorithms. Luniver Press.*
14. Yang, X. S. & Deb, S. (2009). Cuckoo search via LÉVY flights. *Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), India.*
15. Yang, X. S. & Deb, S. (2009). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, Vol. 1, pp. 330–343.
16. American Institute of Steel Construction (AISC), (1989). *Manual of steel construction–allowable stress design.* 9th ed. Chicago: AISC.
17. Valian, E., Mohanna, S & Tavakoli, S. (2011). Improved cuckoo search algorithm for feed forward neural network training. *International Journal of Artificial Intelligence Applications.* Vol. 2, No. 3, pp. 36-43.
18. Walton, S., Hassan, O., Morgan, K. & Brown, M. R. (2011). Modified cuckoo search: A new gradient free optimization algorithm. *Chaos, Solitons and Fractals.* Vol. 44, pp. 710–718.
19. Tuba, M., Subotic, M. & Stanarevic, N. (2011). Modified cuckoo search algorithm for unconstrained optimization problems. *Proceedings of the 5th European Computing Conference (ECC'11)*, pp. 263-268.
20. Camp, C. V. (2007). Design of space trusses using big bang-big crunch optimization. *Journal of Structural Engineering*, Vol. 133, pp. 999-1008.
21. Camp, C. V. & Bichon, B. J. (2004). Design of space trusses using ant colony optimization. *Journal of Structural Engineering*, Vol. 130, pp. 741-751.
22. Kaveh, A. & Talatahari, S. (2008). Ant colony optimization for design of space trusses. *International Journal of Space Structures*, Vol. 23, pp. 167-181.