Advances in Computational Design, *Vol. 1, No. 2 (2016) 119-138*
DOI: http://dx.doi.org/10.12989/acd.2016.1.2.119

Truss optimization with dynamic constraints using UECBO

A. Kaveh* and M. Ilchi Ghazaan

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 16846-13114, Iran

(Received October 16, 2015, Revised November 25, 2015, Accepted November 27, 2015)

Abstract. In this article, hybridization of enhanced colliding bodies optimization (ECBO) with upper bound strategy (UBS) that is called UECBO is proposed for optimum design of truss structures with frequency constraints. The distinct feature of the proposed algorithm is that it requires less computational time while preserving the good accuracy of the ECBO. Four truss structures with frequency limitations selected from the literature are studied to verify the viability of the algorithm. This type of problems is highly non-linear and non-convex. The numerical results show the successful performance of the UECBO algorithm in comparison to the CBO, ECBO and some other metaheuristic optimization methods.

Keywords: optimum design; enhanced colliding bodies optimization; upper bound strategy; truss structures; frequency constraints

1. Introduction

Many metaheuristic algorithms that are mostly inspired by the laws of natural phenomena have been proposed in the last decades as robust tools for dealing with today's optimization problems. Genetic algorithm (GA) (Holland 1975, Goldberg 1989) inspired by Darwin's theory about biological evolutions, particle swarm optimization (PSO) (Kennedy and Eberhart 1995) mimics the social interaction behavior of birds flocking and fish schooling, ant colony optimization (ACO) (Dorigo et al. 1996) simulates the foraging behavior of real life of ant colonies that can establish a shortest route from food source to their nest and vice versa, artificial bee colony (ABC) (Karaboga and Basturk 2007) simulates the intelligent foraging behavior of the honey bees, firefly algorithm (FA) (Yang 2010) is based on the flashing patterns and behaviors of fireflies, cuckoo search (CS) (Yang and Deb 2010) inspired from parasitic breeding behavior of some cuckoo species that lay their eggs in the nests of host birds of other species, ray optimization (RO) (Kaveh and Khayatazad 2012) simulates a set of rays of light passing through a boundary between two transparent materials, colliding bodies optimization (CBO) (Kaveh and Mahdavi 2014) inspired by a collision between two objects in one-dimension, and ant lion optimizer (ALO) (Mirjalili 2015) mimics the hunting mechanism of antlions in nature are some of the algorithms in this field. In recent years, many researches utilized metaheuristic algorithms in engineering problems (Degertekin 2008, Li *et al*. 2010, Sheikhi and Ghoddosian 2013, Tang *et al*. 2013, Kaveh 2014,

*Corresponding author, Professor, E-mail: alikaveh@iust.ac.ir

http://www.techno-press.com/journals/acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

Kaveh *et al*. 2015, Kazemzadeh and Hasançebi 2015, Nigdeli *et al*. 2015).

To avoid the resonance phenomenon and to improve the performance of the structure, some limitations should be imposed on the natural frequency ranges. Optimum design of structures considering natural frequency constraints is described as a highly nonlinear optimization problem with non-convex solution space and multiple local minima which make the challenge of finding the global optimum hard. This kind of problems has been studied since the 1980s (Bellagamba and Yang 1981) and approached with mathematical programming and meta-heuristic algorithms. Grandhi and Venkayya (1989) utilized an optimality criterion based on uniform Lagrangian density for resizing and scaling procedure to locate the constraint boundary, Sedaghati (2005) utilized a new approach using combined mathematical programming based on the sequential quadratic programming (SQP) technique and a finite element solver based on the integrated force method. Lingyun *et al*. (2005) combined the simplex search method and the niche genetic hybrid algorithm (NGHA) for mass minimization of structures with frequency constraints. Gomes (2011) used the particle swarm optimization (PSO) algorithm to study simultaneous layout and sizing optimization of truss structures with multiple frequency constraints. Miguel and Fadel (2012) employed Harmony Search (HS) and Firefly Algorithm (FA), to solve this type of problems. Kaveh and Zolghadr (2012) combined Charged-System Search and Big Bang with trap recognition capability (CSS-BBBC) to solve layout and sizing optimization problems of truss structures with natural frequency constraints. Kaveh and Ilchi Ghazaan (2014b) employed enhanced colliding bodies optimization (ECBO) to study truss optimization with frequency constraints.

The ECBO algorithm is a multi-agent method and the governing laws of collision from the physics are the base of this technique, where these laws determine the movement process of the agents. This algorithm with modification in mass function is combined with the upper bound strategy (Kazemzadeh *et al*. 2013) and proposed for optimum design of trusses with frequency constraints. The new algorithm is called UECBO and is tested by solving four truss weight minimization problems. UECBO is compared to the CBO, ECBO and some previously reported results in the literature. Optimization results demonstrate the efficiency of the proposed algorithm that outperforms its rival.

The rest of the paper is structured as follows. Statement of the optimization design problem with frequency constraints is formulated in Section 2. Section 3 describes the UECBO algorithm besides a brief introduction to ECBO and UBS methods. Four numerical examples are studied in Section 4 in order to show the capability of the proposed algorithm. Finally, in Section 5 some concluding remarks are provided.

2. Problem statements

The paper aims is to minimize the weight of the structure under frequency constraints. The optimization problem can be stated mathematically as follows

$$\begin{aligned} &\text{Find} && \{X\}=[x_1,x_2,..,x_{ng}] \\ &\text{to minimize} && W(\{X\})=\sum_{i=1}^{nm}\rho_i A_i L_i \\ &\text{subjected to:} && \begin{cases}\omega_j \le \omega_j^* \\ \omega_k \ge \omega_k^* \\ x_{i\min} \le x_i \le x_{i\max}\end{cases} \end{aligned} \tag{1}$$

where $\{X\}$ is the vector containing the design variables including cross-sectional areas; ng is the number of design variables; $W(\{X\})$ is the weight of the structure; *nm* is the number of elements of the structure; ρ_i, A_i and L_i denote the material density, cross-sectional area and the length of the *i*th member, respectively; ω_j is the *j*th natural frequency of the structure and ω_j^* is its upper bound; ω_k is the *k*th natural frequency of the structure and ω_k^* is its lower bound; x_{imin} and x_{imax} are the lower and upper bounds of the design variable x_i, respectively.

To handle the constraints, the well-known penalty approach is employed (Kaveh and Talatahari 2012). Thus, the objective function is redefined as follows

$$f(\{X\}) = (1+\varepsilon_1.\upsilon)^{\varepsilon_2} \times W(\{X\}) \tag{2}$$

where υ denotes the sum of the violations of the design constraints. 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 and ε_2 is set to 1.5 and ultimately increased to 3. Such a scheme penalizes the unfeasible solutions more severely as the optimization process proceeds. As a result, in the early stages the agents are free to explore the search space, but at the end they tend to choose solutions with no violation.

3. Optimization algorithms

3.1 The ECBO algorithm

Colliding bodies optimization (CBO) is a meta-heuristic search algorithm that is developed by Kaveh and Mahdavi (2014). CBO is based on the governing laws of one dimensional collision between two bodies from the physics. To improve the performance of CBO, enhanced colliding bodies optimization (ECBO) is introduced by Kaveh and Ilchi Ghazaan (2014a).

In this multi-agent method, each solution vector is considered as a colliding body (CB) and one object collides with another and after a collision of two CBs, which have specified masses and velocities these are separated and moved to new positions with new velocities. Mass of each colliding body X_i defined as

$$m_k = \frac{\dfrac{1}{fit(k)}}{\dfrac{1}{\sum_{i=1}^{n}\dfrac{1}{fit(i)}}}, \qquad k = 1,2,...,n \tag{3}$$

where $fit(i)$ represents the objective function value of the *i*th CB, and *n* is the number of colliding bodies.

In ECBO, colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values. In each iteration, solution vectors which are saved in CM are added to the population and the same number of current worst CBs are deleted. To select the pairs of objects for collision, CBs are sorted according to their objective function values in an increasing order and after that they are divided into two main equal groups: (i) stationary group, (ii) moving group (Fig. 1). The moving objects move to follow stationary objects

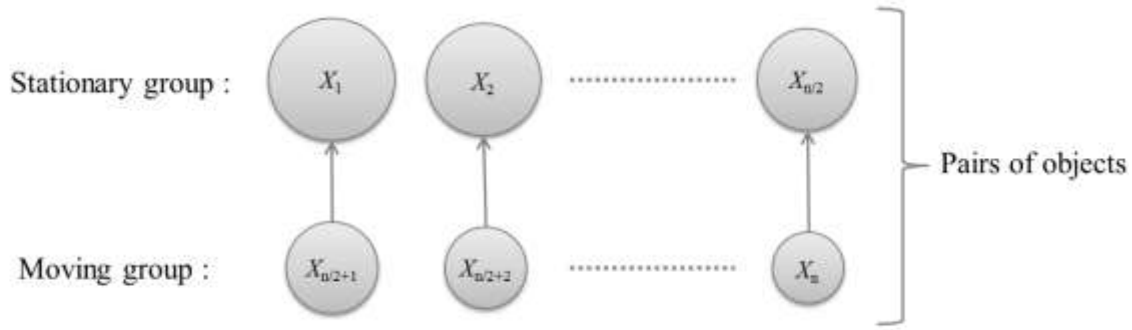

Fig. 1 The pairs of CBs for collision

and a collision occurs between pairs of objects.

The velocity of the stationary bodies before collision is zero

$$v_i = 0, \qquad i = 1,2,...,\frac{n}{2} \tag{4}$$

And also the velocity of moving bodies before collision are given by

$$v_i = x_j - x_i, \qquad i = \frac{n}{2}+1, \frac{n}{2}+2,..., n \qquad j = i - \frac{n}{2} \tag{5}$$

The velocity of each stationary CB after the collision (v'_i) is stated as

$$v_i^{'} = \frac{(m_j + \varepsilon m_j)v_j}{m_i + m_j} \qquad i = 1,2,...,\frac{n}{2} \qquad j = i + \frac{n}{2} \tag{6}$$

Also, the velocity of each moving CB after the collision (v'_i) is defined by

$$v_i^{'} = \frac{(m_i - \varepsilon m_j)v_i}{m_i + m_j} \qquad i = \frac{n}{2}+1, \frac{n}{2}+2,..., n \qquad j = i - \frac{n}{2} \tag{7}$$

Here, ε is the coefficient of restitution (COR) and it is decreases linearly from unit to zero

$$\varepsilon = 1 - \frac{iter}{iter_{\max}} \tag{8}$$

where *iter* is the current iteration number and $iter_{max}$ is the total number of iterations of the optimization process.

New positions of CBs are updated according to their velocities after the collision and the positions of stationary CBs. Therefore, the new position of each stationary CB is defined by

$$x_i^{new} = x_i + rand \circ v_i^{'}, \qquad i = 1,2,...,\frac{n}{2} \tag{9}$$

where x_i^{new}, x_i and v'_i are the new position, previous position and the velocity after the collision of the *i*th CB, respectively. *rand* is a random vector uniformly distributed in the range of [-1,1] and the sign "∘" denotes an element-by-element multiplication. Also, the new position of each moving CB is calculated by

$$x_i^{new} = x_j + rand \circ v_i^{'}, \qquad i = \frac{n}{2}+1, \frac{n}{2}+2, ..., n \qquad j = i - \frac{n}{2} \tag{10}$$

In ECBO, a parameter like ***Pro*** within (0, 1) is introduced and specified whether a component of each CB must be changed or not. For each colliding body ***Pro*** is compared with rn_i (i=1,2,…,n) which is a random number uniformly distributed within (0, 1). If rn_i<***pro***, one design variable of the *i*th CB is selected in random and its value regenerated by

$$x_{ij} = x_{j,\min} + random(x_{j,\max} - x_{j,\min}) \tag{11}$$

where x_{ij} is the *j*th design variable of the *i*th CB. $x_{j,min}$ and $x_{j,max}$ are the minimum and maximum limits of the *j*th design variable.

The optimization process is terminated after a fixed number of iterations.

3.2 Upper bound strategy (UBS)

UBS has been developed recently by Kazemzadeh *et al*. (2013) as a simple, yet an efficient strategy, to reduce the total number of structural analyses through avoiding unnecessary analyses during the course of optimization. The key issue in the UBS is to detect those candidate designs which have no chance to surpass the best design found so far during the iterations of the optimum design process. After identifying those non-improving designs, they are directly excluded from the structural analysis stage, resulting in a significant saving in the computational effort. The current best design can usually be considered as the upper bound for the forthcoming candidates to eliminate unnecessary structural analysis and associated fitness computations for those candidates that have no chance of surpassing the best solution. Basically, the key feature in this approach is to define the penalized weight of the current best solution found during the previous iterations as an upper bound for the net weight of the newly generated candidate solutions. Thus, any new candidate solution with a net weight greater than this upper bound will not be analyzed and this will reduce the computational burden of the optimization algorithm (Kazemzadeh *et al*. 2013).

3.3 The UECBO algorithm

In this article, a new algorithm that is called UECBO is proposed for optimum design of truss structures with dynamic constraints. The ECBO mass function is changed and it is hybridized with UBS leading to this new algorithm (Kaveh and Ilchi 2015). The following steps introduce the UECBO and its pseudo code is provided in Fig. 2.

Step 1: Initialization

The initial locations of CBs are created randomly in search space. The objective function is evaluated for each CB.

Step 2: Saving

Colliding memory (CM) is considered to save some historically best CB vectors and their related objective function values. At each iteration, solution vectors that are saved in the CM are added to the population and the same number of the current worst CBs are deleted. In this

```
procedure UECBO
    UECBO parameters are set
    for all CBs
        Initial location is created randomly
        The values of objective function is evaluated
    end for
    While maximum iterations is not fulfilled
        Colliding memory (CM) is updated
        The upper bound is selected
        The population is updated
        Stationary and moving groups are created
        for all CBs
            The velocity before collision is evaluated by Eqs. (4-5)
            The velocity after collision is evaluated by Eqs. (6-7)
            New location is updated by Eqs. (9-11)
            Net weight is calculated
            If net weight < upper bound
                Penalized weight is calculated
            end if
        end for
    end while
end procedure
```

Fig. 2 Pseudo code of the UECBO algorithm

research, the size of colliding memory is taken as 10% of the population size. Moreover, the objective function value of the so far best design is selected as the upper bound.

Step 3: Creating groups

CBs are sorted according to their objective function values in an increasing order. To select the pairs of CBs for collision, they are divided into two equal groups: (i) stationary group, (ii) moving group.

Step 4: Defining mass

The value of mass for each stationary body is set to C_1 ($0.5 \le C_1 < 1$). Mass of each moving CB is defined by

$$C_2 = 1 - C_1 \tag{12}$$

In this paper, C_1 is set to 0.5.

Step 5: Criteria before the collision

The velocities of stationary and moving bodies before collision are evaluated by Eq. (4) and Eq. (5), respectively.

Step 6: Criteria after the collision

The velocities of stationary and moving bodies after collision are calculated by Eq. (6) and Eq. (7), respectively.

Step 7: Updating CBs

The new locations of CBs are evaluated by Eqs. (9)-(11).

Step 8: CBs evaluation

Net weight of each CB is computed based on Eq. (1) (note that it is not necessary to consider design constraints). Any CB with a net weight greater than upper bound will not be analyzed and its net weight is considered as its objective function value. Other CBs should be analyzed and their objective function values should be calculated based on Eq. (2).

Step 9: Terminal condition check

After the predefined maximum evaluation number, the optimization process is terminated. However, any other terminal condition can be utilized. Here, 20,000 evaluations are considered as maximum function evaluations.

4. Numerical examples

Four truss structures are optimized for minimum weight with the cross-sectional areas of the members being the design variables to verify the efficiency of the UECBO. A population of 40 CBs is used for the first and second examples and 30 CBs are utilized for the rest of problems. For all examples 20 independent optimization runs are carried out as meta-heuristic algorithms have stochastic nature and their performance may be sensitive to initial population.

The algorithms are coded in MATLAB and the structures are analyzed using the direct stiffness method by our own codes. All experiments are carried out on a PC with a Core 2 Duo CPU E4600 running at 2.40 GHz with 2GB of RAM (note that only a single processor is used due to the sequential implementation of the algorithm). The operation system is Windows 7 and version of MATLAB is R2010a.

4.1 A 10-bar plane truss

Schematic of the 10-bar plane truss shown in Fig. 3 is the first test problem. The cross-sectional area of each of the members is considered to be an independent variable. The modulus of elasticity is 68.95 GPa and the material density is 2767.99 kg/m^3 for all elements. At each free node (1-4), a non-structural mass of 453.6 kg is attached. The minimum cross-sectional area of all members is 0.645 cm^2 and the maximum cross-sectional area is taken as 50 cm^2. The first three natural frequencies of the structure must satisfy the following limitations ($f_1 \geq 7$ Hz, $f_2 \geq 15$ Hz and $f_3 \geq 20$ Hz).

This example has been studied by Wang *et al*. (2004) utilizing an evolutionary node shift method, Lingyun *et al*. (2005) using a niche hybrid genetic algorithm, Gomes (2011) employing the particle swarm optimization (PSO) algorithm. Miguel and Fadel (2012) have investigated this problem using the firefly algorithm (FA). Kaveh and Zolghadr (2014) utilized democratic particle swarm optimization (DPSO) to optimize this structure. Kaveh and Ilchi (2014b) employed CBO and ECBO algorithms to study this example.

Table 1 represents results obtained by the various methods and Table 2 shows corresponding natural frequencies. The lightest design (i.e., 531.05 kg) is obtained by UECBO algorithm. Fig. 4 shows the convergence curves of the best results obtained by CBO, ECBO and UECBO. The best designs have been located at 5000, 6300 and 4,932 analyses for CBO, ECBO and UECBO algorithms, respectively. The average require computational time, respectively, are 5.59, 5.72 and 6.06 s. The amount of saving in structural analyses at each iteration of the UECBO shown in Fig. 5.

Table 1 Optimal design comparison for the 10-bar plane truss

Design variable	Areas (cm^2)							
	Wang *et al.* (2004)	Lingyun *et al.* (2005)	Gomes (2011)	Miguel and Fadel (2012)	Kaveh and Zolghadr (2014)	Kaveh and Ilchi (2014b)		Present work
						CBO	ECBO	UECBO
1	32.456	42.234	37.712	36.198	35.944	36.6281	34.9457	35.2759
2	16.577	18.555	9.959	14.030	15.530	15.9742	14.1340	14.1247
3	32.456	38.851	40.265	34.754	35.285	34.9146	35.5134	35.2198
4	16.577	11.222	16.788	14.900	15.385	14.0328	14.3854	15.3591
5	2.115	4.783	11.576	0.654	0.648	0.6450	0.645	0.6450
6	4.467	4.451	3.955	4.672	4.583	4.6117	4.6889	4.6446
7	22.810	21.049	25.308	23.467	23.610	26.0932	24.3026	22.7704
8	22.810	20.949	21.613	25.508	23.599	21.7484	24.9174	25.5137
9	17.490	10.257	11.576	12.707	13.135	12.0427	12.8177	13.3722
10	17.490	14.342	11.186	12.351	12.357	13.0782	12.5752	12.2684
Weight (kg)	553.8	542.75	537.98	531.28	532.39	531.50	531.09	531.05
Average optimized weight (Kg)	N/A	552.447	540.89	535.07	537.80	536.09	535.91	535.30
Standard deviation on average weight (kg)	N/A	4.864	6.84	3.64	4.02	3.85	3.29	3.02
Average computational time (s)	-	-	-	-	-	5.59	5.72	6.06

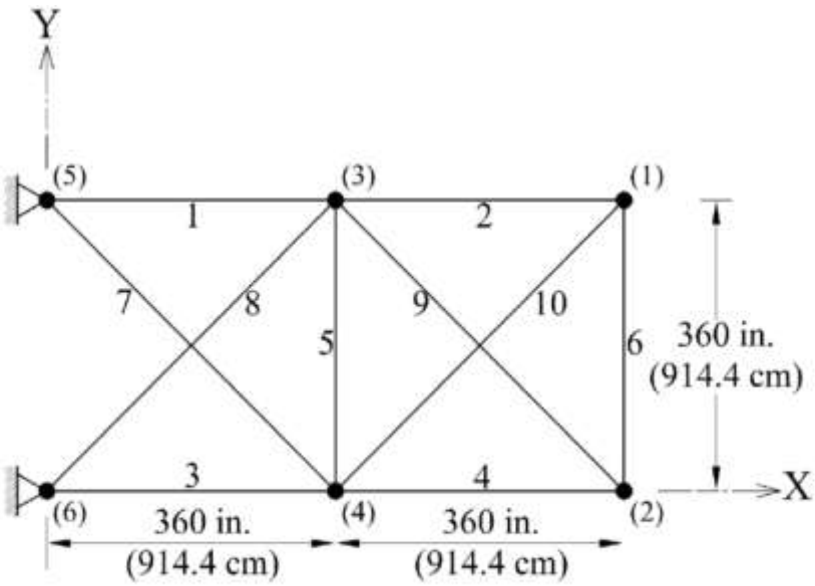

Fig. 3 Schematic of the 10-bar plane truss

Table 2 Optimal design of the natural frequencies (Hz) (the 10-bar plane truss)

Frequency number	**Natural frequencies (Hz)**							
	Wang *et al.* (2004)	Lingyun *et al.* (2005)	Gomes (2011)	Miguel and Fadel (2012)	Kaveh and Zolghadr (2014)	Kaveh and Ilchi (2014b)		Present work
						CBO	ECBO	UECBO
1	7.011	7.008	7.000	7.0002	7.000	7.000	7.000	7.000
2	17.302	18.148	17.786	16.1640	16.187	16.136	16.127	16.124
3	20.001	20.000	20.000	20.0029	20.000	20.000	20.001	20.000
4	20.100	20.508	20.063	20.0221	20.021	20.000	20.004	20.001
5	30.869	27.797	27.776	28.5428	28.470	28.216	28.676	28.422
6	32.666	31.281	30.939	28.9220	29.243	29.295	28.969	29.365
7	48.282	48.304	47.297	48.3538	48.769	48.544	48.179	48.379
8	52.306	53.306	52.286	50.8004	51.389	51.302	50.658	50.966

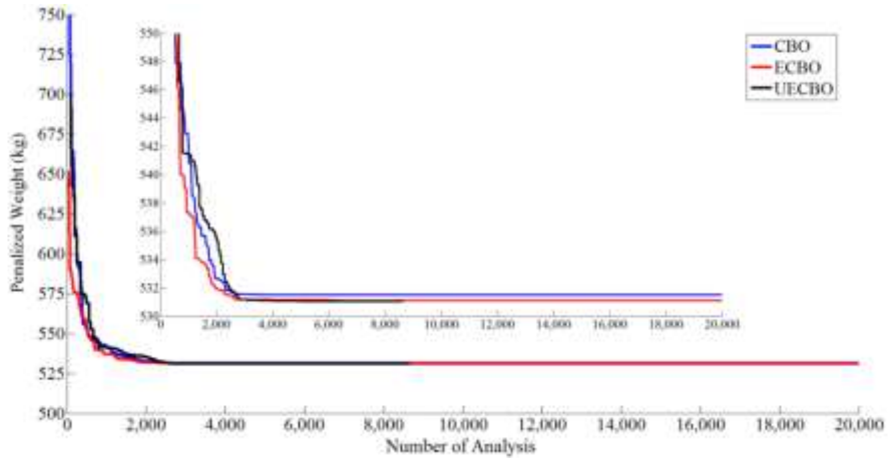

Fig. 4 The convergence curve for the 10-bar plane truss

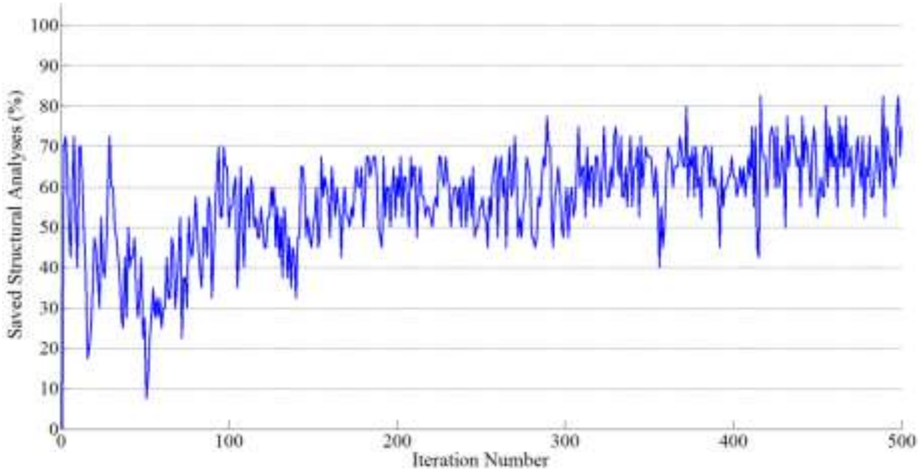

Fig. 5 Saving in structural analyses using the UECBO algorithm in the 10-bar plane truss problem

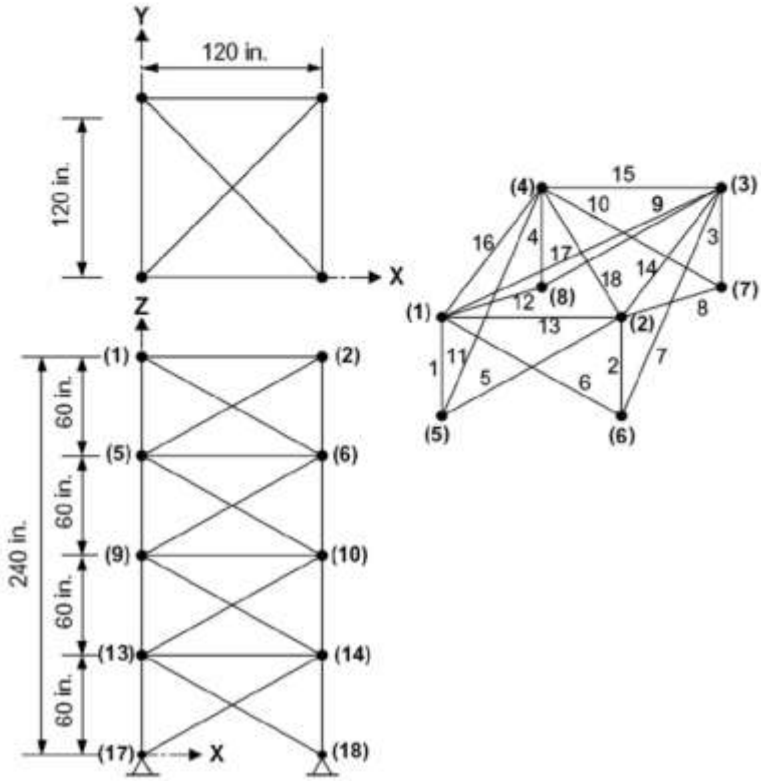

Fig. 6 Schematic of the spatial 72-bar truss

4.2 A 72-bar space truss

Fig. 6 shows the schematic of the 72-bar space truss. The 72 members are categorized in sixteen design groups, because of symmetry. The elastic modulus is 68.95 GPa and the material density is 2767.99 kg/m^3 for all elements. Four non-structural masses of 2268 kg are attached to the nodes 1 through 4. The allowable minimum cross-sectional area of all elements is set to 0.645 cm^2. This example has two frequency constraints. The first frequency is required to be f_1=4 Hz and the third frequency is required to be $f_3 \geq 6$ Hz.

This example is a well-known benchmark problem in the field of frequency constraint structural optimization and has been investigated by different researchers: Gomes (2011) used PSO, Miguel and Fadel (2012) utilized FA and Kaveh and Ilchi (2014b) employed CBO and ECBO.

The design vectors, the weight of the corresponding structures and statistical information of the solution obtained by different researchers is presented in Table 3. The ECBO yields the least weight for this example, which is 327.648 kg. Table 4 reports the natural frequencies of the optimized structures and it is clear that none of the frequency constraints are violated. Fig. 7 provides the convergence rates of the best result found by the CBO, ECBO and UECBO. They, respectively, found the optimum weight after 4000, 14800 and 6548 analyses. Their average computational times are 48.96, 50.01 and 36.37 s, respectively. Fig. 8 shows the amount of saving in structural analyses at each iteration of the UECBO.

Table 3 Optimal design comparison for the 72-bar space truss

Element group	**Members in the group**	**Areas (cm^2)**				
		Gomes (2011)	Miguel and Fadel (2012)	Kaveh and Ilchi (2014b)		Present work
				CBO	ECBO	UECBO
1	**1-4**	2.987	3.3411	3.7336	3.5498	3.5199
2	**5-12**	7.849	7.7587	7.9355	7.8356	7.8832
3	**13-16**	0.645	0.6450	0.6450	0.645	0.6451
4	**17-18**	0.645	0.6450	0.6450	0.645	0.6450
5	**19-22**	8.765	9.0202	8.3765	8.1183	8.1334
6	**23-30**	8.153	8.2567	8.0889	8.1338	8.0073
7	**31-34**	0.645	0.6450	0.6450	0.645	0.6450
8	**35-36**	0.645	0.6450	0.6450	0.6450	0.6453
9	**37-40**	13.450	12.0450	12.9491	12.6231	12.8119
10	**41-48**	8.073	8.0401	8.0524	8.0971	8.1172
11	**49-52**	0.645	0.6450	0.6450	0.6450	0.6450
12	**53-54**	0.645	0.6450	0.6450	0.645	0.6450
13	**55-58**	16.684	17.3800	16.6629	17.3908	17.2088
14	**59-66**	8.159	8.0561	8.0557	8.0634	8.1232
15	**67-70**	0.645	0.6450	0.645	0.645	0.6450
16	**71-72**	0.645	0.6450	0.645	0.645	0.6450
Weight (kg)		328.823	327.691	327.740	327.653	327.648
Average optimized weight (Kg)		332.24	329..89	328.20	327.76	327.73
Standard deviation on average weight (kg)		4.23	2.59	0.54	0.06	0.07
Average computational time (s)		-	-	48.96	50.01	36.37

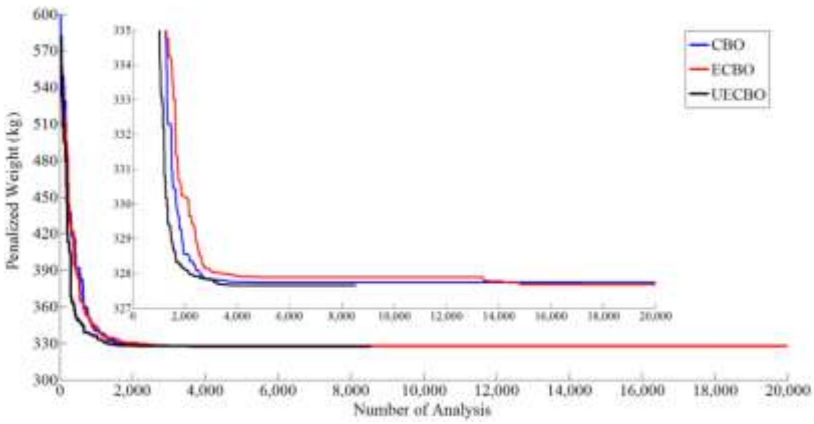

Fig. 7 The convergence curve for the 72-bar space truss

Table 4 Optimal design of the natural frequencies (Hz) (the 72-bar space truss)

Frequency number	Natural frequencies (Hz)				
	Gomes (2011)	Miguel and Fadel (2012)	Kaveh and Ilchi (2014b)		Present work
			CBO	ECBO	UECBO
1	4.000	4.0000	4.000	4.000	4.000
2	4.000	4.0000	4.000	4.000	4.000
3	6.000	6.0000	6.000	6.000	6.000
4	6.219	6.2468	6.267	6.246	6.246
5	8.976	9.0380	9.101	9.071	9.068

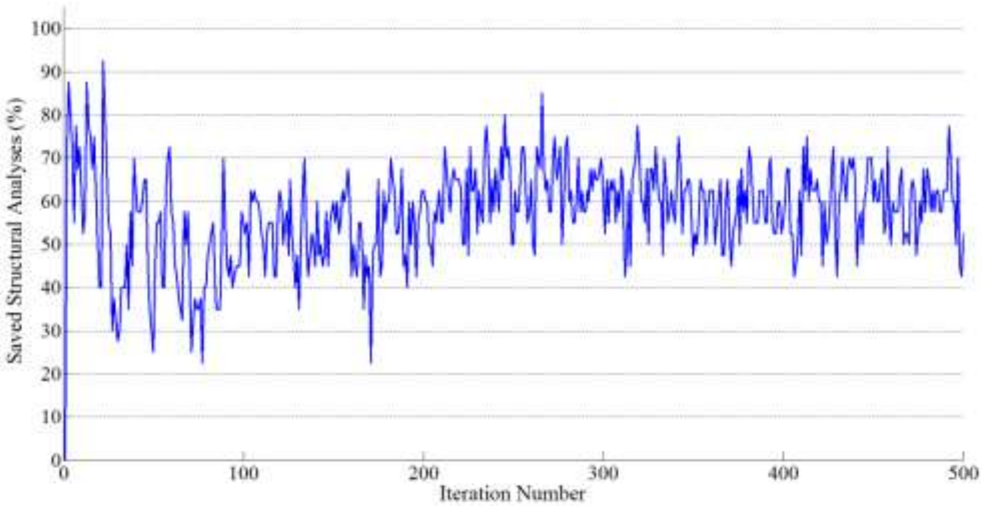

Fig. 8 Saving in structural analyses using the UECBO algorithm in the 72-bar space truss problem

4.3 A 120-bar dome truss

The third test problem regards the 120-bar dome truss depicted in Fig. 9. The modulus of elasticity is 210 GPa and the material density is 7971.810 kg/m^3 for all elements. Non-structural masses are attached to all free nodes as follows: 3000 kg at node one, 500 kg at nodes 2-13 and 100 kg at the remaining nodes. The symmetry of the structure about *x*-axis and *y*-axis is considered to group the 120 members into 7 independent size variables. The minimum cross-sectional area of all members is 1 cm^2 and the maximum cross-sectional area is taken as 129.3 cm^2. The frequency constraints are as followings: $f_1 \geq 9$ Hz and $f_2 \geq 11$ Hz.

This example has been studied by Kaveh and Zolghadr (2012) used the hybridized CSS-BBBC with a trap recognition capability to optimize this structure. Also, they have studied this problem using DPSO (Kaveh and Zolghadr 2014). Kaveh and Ilchi Ghazaan (2014b) employed CBO and ECBO to study this example.

Table 5 observes the comparison of the results of utilized algorithm with the outcomes of other algorithms. Frequency constraints are satisfied by all methods (see Table 6). The DPSO yields the least weight for this example, which is 8890.48 kg. The other design weights are 9046.34 kg by

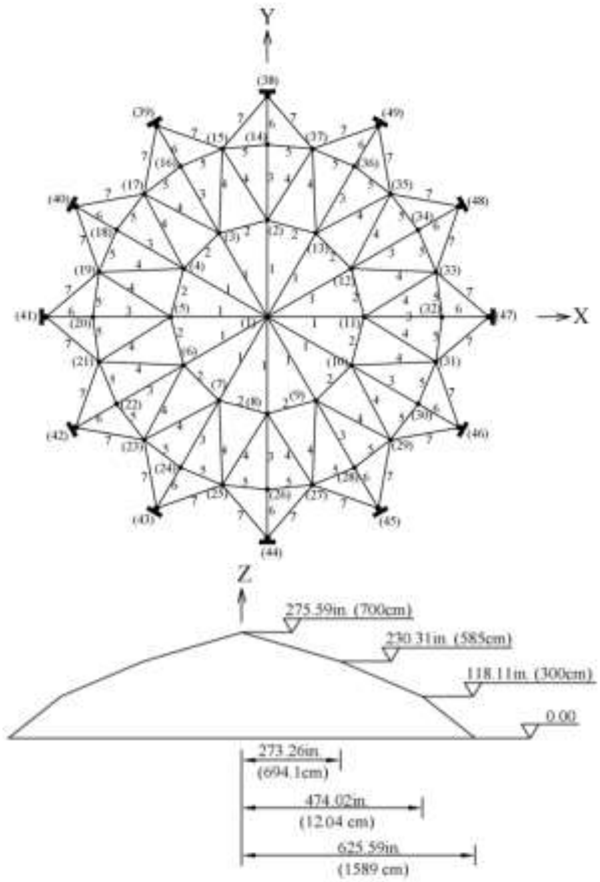

Fig. 9 Schematic of the spatial 120-bar dome truss

CSS-BBBC, 8890.69 kg by CBO, 8896.50 kg by ECBO and 8894.54 kg by UECBO algorithm. Fig. 10 compares the best runs convergence histories for the CBO, ECBO and UECBO. The best designs have been found at 3700, 7700 and 3455 analyses for CBO, ECBO and UECBO algorithms, respectively. Moreover, the average require computational time, respectively, are 242.69, 244.85 and 125.23 s. The amount of saving in structural analyses at each iteration of the UECBO shown in Fig. 11.

4.4 A 200-bar planar truss

The sizing optimization of a planar 200-bar truss shown in Fig. 12 is the last test case. The elastic modulus is 210 GPa and the material density is 7,860 kg/m^3 for all elements. Non-structural masses of 100 kg are attached to the nodes 1 to 5. The minimum admissible cross-sectional areas are 0.1 cm^2. Because of the symmetry, the bars are categorized into 29 groups. The first three natural frequencies of the structure must satisfy the following limitations ($f_1 \geq 5$ Hz, $f_2 \geq 10$ Hz, $f_3 \geq 15$ Hz).

This truss has been studied using the hybridized CSS-BBBC with a trap recognition capability as a frequency constraint weight optimization problem by Kaveh and Zolghadr (2012). Kaveh and

Table 5 Optimal design comparison for the 120-bar dome truss

Design variable	Areas (cm^2)				
	Kaveh and Zolghadr (2012)	Kaveh and Zolghadr (2014)	Kaveh and Ilchi (2014b)		Present work
			CBO	ECBO	UECBO
1	17.478	19.607	19.7738	19.8290	19.5286
2	49.076	41.290	40.6757	41.4037	40.8324
3	12.365	11.136	11.6056	11.0055	11.5304
4	21.979	21.025	21.4601	21.2971	21.6495
5	11.190	10.060	9.8104	9.4718	10.2915
6	12.590	12.758	12.2866	13.0176	12.6229
7	13.585	15.414	15.1417	15.2840	14.7256
Weight (kg)	9046.34	8890.48	8890.69	8896.50	8894.54
Average optimized weight (Kg)	N/A	8895.99	8945.64	8920.16	8936.9
Standard deviation on average weight (kg)	N/A	4.26	38.33	20.12	29.38
Average computational time (s)	-	-	242.69	244.85	125.23

Table 6 Optimal design of the natural frequencies (Hz) (the 120-bar dome truss)

Frequency number	Natural frequencies (Hz)				
	Kaveh and Zolghadr (2012)	Kaveh and Zolghadr (2014)	Kaveh and Ilchi Ghazaan (2014b)		Present work
			CBO	ECBO	UECBO
1	9.000	9.0001	9.000	9.001	9.001
2	11.007	11.0007	11.000	11.001	11.000
3	11.018	11.0053	11.000	11.003	11.000
4	11.026	11.0129	11.010	11.010	11.010
5	11.048	11.0471	11.049	11.052	11.050

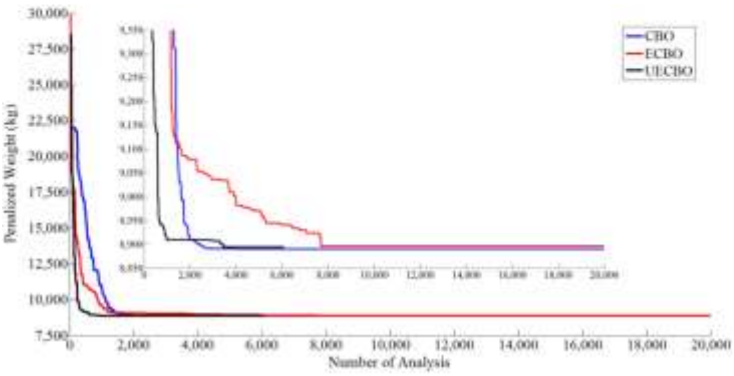

Fig. 10 The convergence curve for the 120-bar dome truss

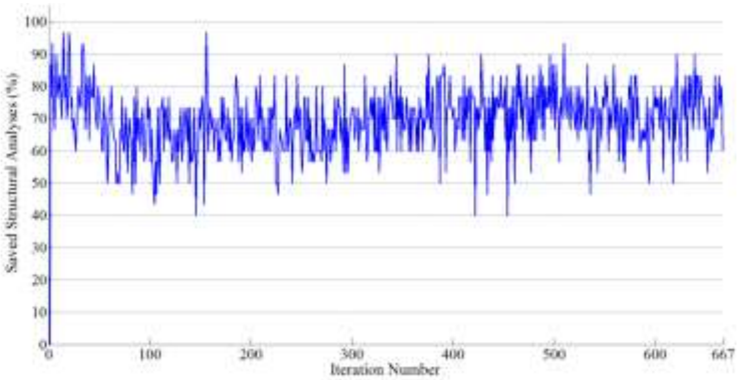

Fig. 11 Saving in structural analyses using the UECBO algorithm in the 120-bar dome truss problem

Table 7 Optimal design comparison for the 200-bar planar truss

Element group	Members in the group	Areas (cm^2)			
		Kaveh and Zolghadr (2012)	Kaveh and Ilchi (2014b)		Present work
			CBO	ECBO	UECBO
1	**1,2,3,4**	0.2934	0.3059	0.2993	0.3002
2	**5,8,11,14,17**	0.5561	0.4476	0.4497	0.4890
3	**19,20,21,22,23,24**	0.2952	0.1000	0.1000	0.1000
4	**18,25,56,63,94,101,132,139,170,177**	0.1970	0.1001	0.1	0.1000
5	**26,29,32,35,38**	0.8340	0.4944	0.5137	0.5277
6	**6,7,9,10,12,13,15,16,27,28,30,31,33, 34,36,37**	0.6455	0.8369	0.7914	0.8310
7	**39,40,41,42**	0.1770	0.1001	0.1013	0.1001
8	**43,46,49,52,55**	1.4796	1.5514	1.4129	1.3841
9	**57,58,59,60,61,62**	0.4497	0.1000	0.1019	0.1000
10	**64,67,70,73,76**	1.4556	1.5286	1.6460	1.5912
11	**44,45,47,48,50,51,53,54,65,66,68,69, 71,72,74,75**	1.2238	1.1547	1.1532	1.1502
12	**77,78,79,80**	0.2739	0.1000	0.1000	0.1008
13	**81,84,87,90,93**	1.9174	2.9980	3.1850	3.0023
14	**95,96,97,98,99,100**	0.1170	0.1017	0.1034	0.1007
15	**102,105,108,111,114**	3.5535	3.2475	3.3126	3.2767
16	**82,83,85,86,88,89,91,92,103,104,106, 107,109,110,112,113**	1.3360	1.5213	1.5920	1.6017
17	**115,116,117,118**	0.6289	0.3996	0.2238	0.2309
18	**119,122,125,128,131**	4.8335	4.7557	5.1227	5.0228
19	**133,134,135,136,137,138**	0.6062	0.1002	0.1050	0.1057
20	**140,143,146,149,152**	5.4393	5.1359	5.3707	5.2667
21	**120,121,123,124,126,127,129,130,141, 142,144,145,147,148,150,151**	1.8435	2.1181	2.0645	2.1287
22	**153,154,155,156**	0.8955	0.9200	0.5443	0.7337

Table 7 Continued

Element group	Members in the group	Areas (cm^2)			
		Kaveh and Zolghadr (2012)	Kaveh and Ilchi (2014b)		Present work
			CBO	ECBO	UECBO
23	**157,160,163,166,169**	8.1759	7.3084	7.6497	7.9257
24	**171,172,173,174,175,176**	0.3209	0.1185	0.1000	0.1000
25	**178,181,184,187,190**	10.98	7.6901	7.6754	8.1735
26	**158,159,161,162,164,165,167,168,179, 180,182,183,185,186,188,189**	2.9489	3.0895	2.7178	2.7758
27	**191,192,193,194**	10.5243	10.6462	10.8141	10.1047
28	**195,197,198,200**	20.4271	20.7190	21.6349	21.2172
29	**196,199**	19.0983	11.7463	10.3520	10.9900
Weight (kg)		2298.61	2161.15	2158.08	2157.65
Average optimized weight (Kg)		N/A	2447.52	2159.93	2161.36
Standard deviation on average weight (kg)		N/A	301.29	1.57	2.72
Average computational time (s)		-	190.60	193.41	87.22

Table 8 Optimal design of the natural frequencies (Hz) (the 200-bar planar truss)

Frequency number	Natural frequencies (Hz)			
	Kaveh and Zolghadr (2012)	Kaveh and Ilchi (2014b)		Present work
		CBO	ECBO	UECBO
1	5.010	5.000	5.000	5.000
2	12.911	12.221	12.189	12.260
3	15.416	15.088	15.048	15.096
4	17.033	16.759	16.643	16.696
5	21.426	21.419	21.342	21.452
6	21.613	21.501	21.382	21.534

Ilchi (2014b) utilized CBO and ECBO algorithms to optimize this problem.

The optimized designs found by the different algorithms are compared in Table 7 that shows also the corresponding structural weights, average optimized weight, standard deviation on average weight and average computational time. UECBO designed the lightest structure overall. Table 8 reports the natural frequencies of the optimized structures and it is clear that none of the frequency constraints are violated. Fig. 13 compares the convergence curves for the best result obtained by CBO, ECBO and UECBO. They, respectively, found the optimum weight after 10500,

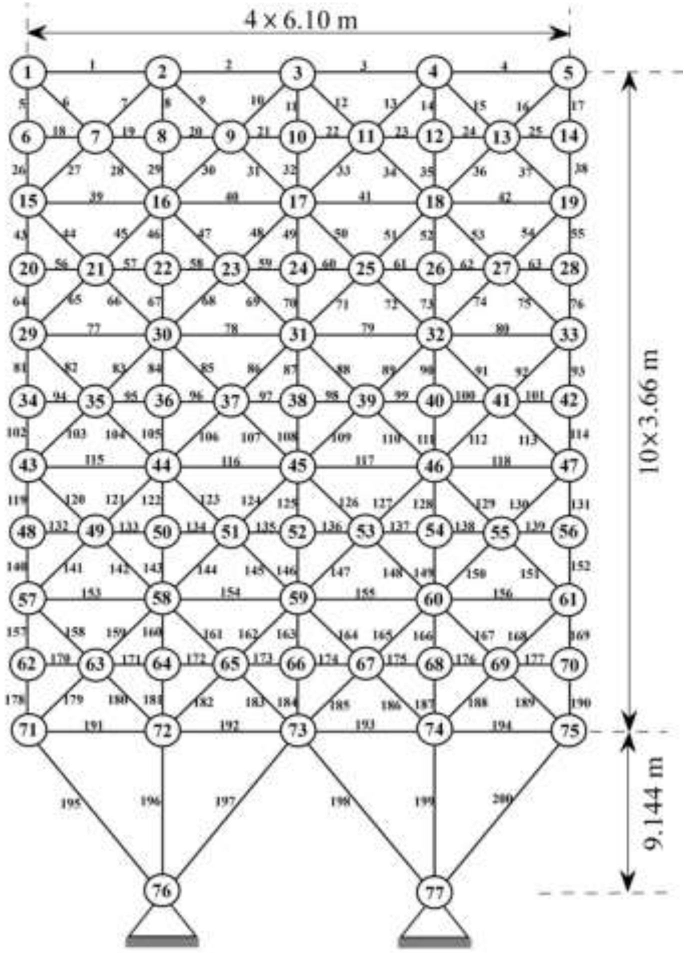

Fig. 12 Schematic of the 200-bar planar truss

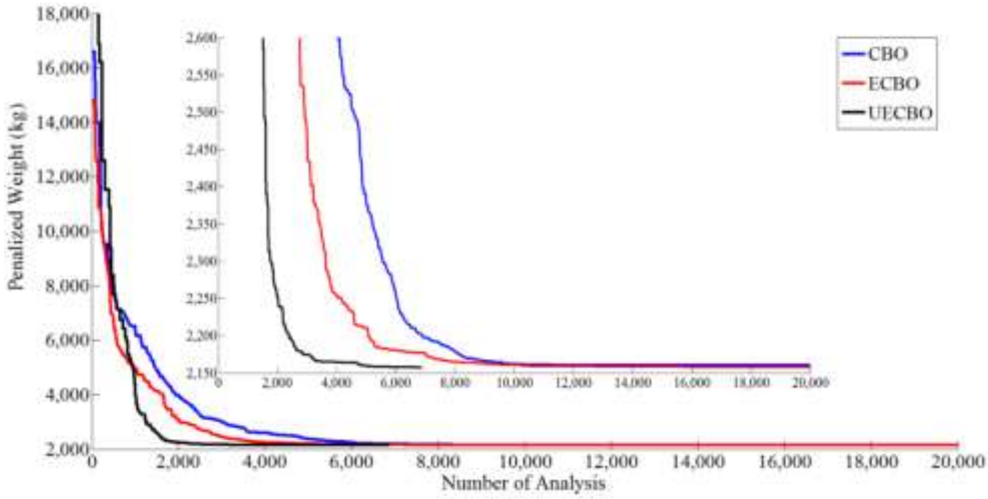

Fig. 13 The convergence curve for the 200-bar planar truss

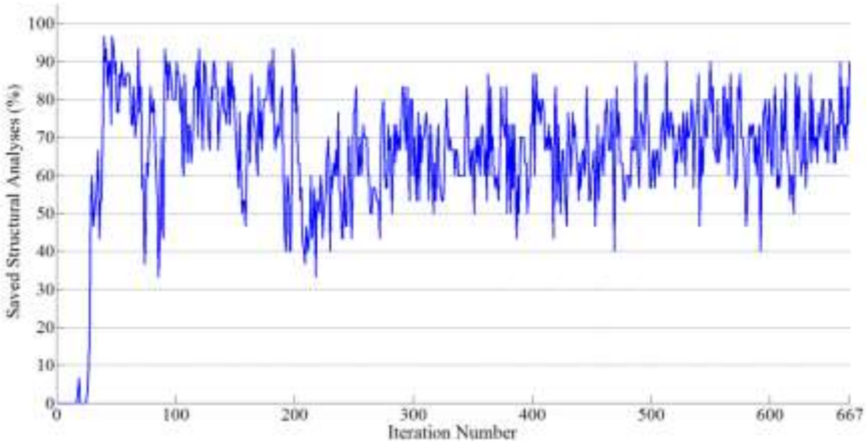

Fig. 14 Saving in structural analyses using the UECBO algorithm in the 200-bar planar truss problem

14700 and 6675 analyses. Their average computational times are 190.60, 193.41 and 87.22 s, respectively. Fig. 14 shows the amount of saving in structural analyses at each iteration of the UECBO.

5. Conclusions

A new hybrid algorithm combining enhanced colliding bodies optimization with upper bound strategy which is called UECBO is proposed in this research. In order to demonstrate the efficiency and robustness of proposed approach, UECBO is applied to four sizing optimization of truss structures with multiple natural frequency constraints. This class of problem is highly non-linear and non-convex dynamic optimization problems since mass reduction conflicts with the frequency constraints. UECBO found the best design in all examples except the third one. The average optimized weight and standard deviation on average weight obtained by UECBO are also suitable. The convergence history curves show that the UECBO is generally faster optimizer than CBO and ECBO. Furthermore, it can be seen from the tables that average computational time required for UECBO are about 26%, 48% and 54% less than CBO and ECBO algorithms in 2nd, 3rd and 4th examples, respectively. To sum up, the comparison shows the superiority of the presented method compared to its rival.

References

Bellagamba, L. and Yang, T.Y. (1981), “Minimum mass truss structures with constraints on fundamental natural frequency”, *AIAA J.*, **19**(11), 1452-1458.

Degertekin, S.O. (2008), “Harmony search algorithm for optimum design of steel frame structures: a comparative study with other optimization methods”, *Struct. Eng. Mech.*, **29**(4), 391-410.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), “The ant system: optimization by a colony of cooperating agents”, *IEEE Trans. Syst. Man Cybern.*, **26**(1), 29-41.

Goldberg, D.E. (1989), *Genetic algorithms in search optimization and machine learning*, Boston: Addison-

Wesley.
Gomes, H.M. (2011), “Truss optimization with dynamic constraints using a particle swarm algorithm”, *Expert Syst. Appl.*, **38**(1), 957-968.
Grandhi, R.V. and Venkayya, V.B. (1988), “Structural optimization with frequency constraints”, *AIAA J.*, **26**(7), 858-66.
Holland, J.H. (1975), *Adaptation in natural and artificial systems*, Ann Arbor: University of Michigan Press.
Karaboga, D. and Basturk, B. (2007), “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, *J. Global Optim.*, **39**(3), 459-71.
Kaveh, A. (2014), *Advances in metaheuristic algorithms for optimal design of structures*, Springer International Publishing, Switzerland.
Kaveh, A., Fahimi-Farzam, M. and Kalateh-Ahani, M. (2015), “Optimum design of steel frame structures considering construction cost and seismic damage”, *Smart Struct. Syst.*, **16**(1), 1-26.
Kaveh, A. and Ilchi Ghazaan, M. (2014a), “Enhanced colliding bodies optimization for design problems with continuous and discrete variables”, *Adv. Eng. Softw.*, **77**, 66-75.
Kaveh, A. and Ilchi Ghazaan, M. (2014b), “Enhanced colliding bodies algorithm for truss optimization with frequency constraints”, *J. Comput. Civ. Eng.*, ASCE, **29**(6), 04014104.
Kaveh, A. and Ilchi Ghazaan, M. (2015), “Hybridization of ECBO and UBS for Optimum design of skeletal structures”, submitted for publication.
Kaveh, A. and Khayatazad, M. (2012), “A new meta-heuristic method: ray optimization”, *Comput. Struct.*, **112**, 283-294.
Kaveh, A. and Mahdavi, V.R. (2014), “Colliding Bodies Optimization method for optimum design of truss structures with continuous variables”, *Adv. Eng. Softw.*, **70**, 1-12.
Kaveh, A. and Talatahari, S. (2012), “Charged system search for optimal design of frame structures”, *Appl. Soft Comput.*, **12**(1), 382-393.
Kaveh, A. and Zolghadr, A. (2012), “Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability”, *Comput. Struct.*, **102**, 14-27.
Kaveh, A. and Zolghadr, A. (2014), “Democratic PSO for truss layout and size optimization with frequency constraints”, *Comput. Struct.*, **130**, 10-21.
Kazemzadeh Azad, S. and Hasançebi, O. (2015), “Computationally efficient discrete sizing of steel frames via guided stochastic search heuristic”, *Comput. Struct.*, **156**, 12-28.
Kazemzadeh Azad, S., Hasançebi, O. and Kazemzadeh Azad, S. (2013), “Upper bound strategy for metaheuristic based design optimization of steel frames”, *Adv. Eng. Softw.*, **57**, 19-32.
Kennedy, J. and Eberhart, R.C. (1995), “Particle swarm optimization”, *Proceedings of the IEEE International Conference Neural Network*, 1942-8.
Li, G., Lu, H. and Liu, X. (2010), “A hybrid simulated annealing and optimality criteria method for optimum design of RC buildings”, *Struct. Eng. Mech.*, **35**(1), 19-35.
Lingyun, W., Mei, Z., Guangming, W. and Guang M. (2005), “Truss optimization on shape and sizing with frequency constraints based on genetic algorithm”, *Comput. Mech.*, **35**(5), 361-368.
Miguel, L.F.F. and Fadel Miguel L.F. (2012), “Shape and size optimization of truss structures considering dynamic constraints through modern meta-heuristic algorithms”, *Expert Syst. Appl.*, **39**(10), 9458-9467.
Mirjalili, S. (2015), “The ant lion optimizer”, *Adv. Eng. Softw.*, **83**, 80-98.
Nigdeli, S.M., Bekdas, G., Kim, S. and Geem, Z.W. (2015), “A novel harmony search based optimization of reinforced concrete biaxially loaded columns”, *Struct. Eng. Mech.*, **54**(6), 1097-1109.
Sedaghati, R. (2005), “Benchmark case studies in structural design optimization using the force method”, *Int. J. Solids Struct.*, **42**(21), 5848-5871.
Sheikhi, M. and Ghoddosian, A. (2013), “A hybrid imperialist competitive ant colony algorithm for optimum geometry design of frame structures”, *Struct. Eng. Mech.*, **46**(3), 403-416.
Tang, H., Zhang, W., Xie, L. and Xue, S. (2013), “Multi-stage approach for structural damage identification using particle swarm optimization”, *Smart Struct. Syst.*, **11**(1), 69-86.
Wang, D., Zhang, W.H., and Jiang, J.S. (2004), “Truss optimization on shape and sizing with frequency constraints”, *AIAA J.*, **42**(3), 622-630.

Yang, X.S. (2010), “Firefly algorithm, stochastic test functions and design optimization”, *Int. J. Bio-Inspired Comput.*, **2**(2), 78-84.

Yang, X.S. and Deb, S. (2010), “Engineering optimisation by cuckoo search”, *Int. J. Math. Model. Numer. Optim.*, **1**(4), 330-343.

CC