INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2019; 9(3):499-523

STATIC AND DYNAMIC OPPOSITION-BASED LEARNING FOR COLLIDING BODIES OPTIMIZATION

M. Shahrouzi*, †, A. Barzigar, D. Rezazadeh
Civil Engineering Department, Faculty of Engineering, Kharazmi University, Tehran, Iran

ABSTRACT

Opposition-based learning was first introduced as a solution for machine learning; however, it is being extended to other artificial intelligence and soft computing fields including meta-heuristic optimization. It not only utilizes an estimate of a solution but also enters its counter-part information into the search process. The present work applies such an approach to *Colliding Bodies Optimization* as a powerful meta-heuristic with several engineering applications. Special combination of static and dynamic opposition-based operators are hybridized with CBO so that its performance is enhanced. The proposed OCBO is validated in a variety of benchmark test functions in addition to structural optimization and optimal clustering. According to the results, the proposed method of opposition-based learning has been quite effective in performance enhancement of parameter-less colliding bodies optimization.

Keywords: Opposition-based learning; truss structure; building frame; sizing design; geometry optimization; ground motion clustering.

Received: 2 November 2018; Accepted: 25 February 2019

1. INTRODUCTION

Application of meta-heuristic algorithms is growing fast for engineering problems. These methods can be categorized into bio-inspired, physics-inspired, cultural and chemical-inspired algorithms [1]. They are desired choices for dealing with discontinuous, multimodal, non-smooth and non-convex functions. Some of the popular meta-heuristics are Harmony Search [2, 3], Opposition-Switching Search [4], Charged System Search [5], Pseudo-random Directional Search [6], Tug of War Optimization [7], Colliding Bodies Optimization [8, 9], Water Evaporation Optimization [10], Observer-Teacher-Learner-Based Optimization [11], Stochastic Directional Search [12] and Vibrating Particles Search [13]

*Corresponding author: Faculty of Engineering, Kharazmi University, Tehran, Iran
†E-mail address: shahruzi@khu.ac.ir (M. Shahrouzi)

among several others. Every such algorithm applies its own operators to exploite memory or explore the design space [14].

The concept of *Opposition-Based Learning* (OBL) was introduced by Tizhoosh at 2005 [15]. According to OBL, for every candidate solution X, its opposite $X^{Opp.}$ is simultaneously considered as an extra search point to capture the problem optimum. In another word, OBL theory indicates that taking into account both X_i and $X_i^{Opp.}$ reveals better approximation of the problem solution than just trusting on X_i. OBL has already been applied to a number of meta-heuristics including Differential Evolution [16], Harmony Search [17], Gravitational Search Algorithm [18] and Opposition-Switching Search [4].

A variety of definitions for the opposite of solution and implementation strategies can be found in literature for applying OBL to the search algorithms [19]. According to a common impelemenation strategy, the selection pool in every iteration is extended to include not only the main population of agents but also their opposite solutions. In such a case, quality enhancement of the resulted optimum is preserved or even mathematically proven, however, the strategy doubles the required number of fitness calls in every iteration.

In this paper, a more efficient OBL strategy is offered to accelerate the convergence rate of Colliding Bodies Optimization. The proposed *Opposition-based Colliding Bodies Optimization*, OCBO, utilizes two types of OBL; i.e. static and dynamic opposition of colliding bodies. The rest of this paper is organized as follows: CBO basics are reviewed in Section 2. In Section 3, the concept of OBL is briefly explained and the proposed algorithm is presented in Section 4. A comprehensive set of experimental results for unconstrained and constrained problems are provided in Sections 4 and 5. Concluding remarks are finally discussed in Section 6.

2. COLLIDING BODIES OPTIMIZATION

2.1 Collision principals

Collision between bodies are governed by the laws of momentum and energy. When a collision occurs in an isolated system, the total momentum is conserved; i.e. momentum of all the system objects remains constant before and after the collision provided that there are no net external forces acting upon them. Conservation of the total momentum for two colliding bodies can be expressed by the following equation:

$$m_1 v_1 + m_2 v_2 = m_1 v_1^{'} + m_2 v_2^{'} \tag{1}$$

Conservation of the total kinetic energy is expressed as:

$$\frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_1 v_1'^2 + \frac{1}{2} m_2 v_2'^2 + Q \tag{2}$$

where v_1 and v_2 are the initial velocities of the first and the second objects before impact, respectively. After impact $v_1^{'}$ denotes final velocity of the first object and $v_2^{'}$ stands for final velocity of the second object. Masses of the first and the second objects are denoted by m_1 and m_2, respectively. The term Q stands for the kinetic energy loss due to the impact [8]. Consequent velocities after one-dimensional collision are calculated by:

$$v_1^{'} = \frac{\left(m_1 - \varepsilon m_2\right)v_1 + \left(m_2 + \varepsilon m_2\right)v_2}{m_1 + m_2} \tag{3}$$

$$v_2^{'} = \frac{\left(m_2 - \varepsilon m_1\right)v_2 + \left(m_1 + \varepsilon m_1\right)v_1}{m_1 + m_2} \tag{4}$$

where ε; stands for the *Coefficient Of Restitution* (COR) between two colliding bodies. It is defined as the ratio relative velocity of separation over the relative velocity of approach:

$$\varepsilon = \left|\frac{v_2^{'} - v_1^{'}}{v_2 - v_1}\right| = \frac{v'}{v} \tag{5}$$

Such a coefficient of restitution ε varies between 0 and 1 as the collision state varies between the following conditions:

(1)Perfect elastic collision; in which there is no loss of kinetic energy i.e. Q=0 and thus $\varepsilon = 1$.

(2)Perfect inelastic collision; that occurs when two bodies does not get farther from each other after they collide. It means their relative velocity and consequently the coefficient of restitution ε equals zero.

2.2 Colliding bodies optimization

Colliding Bodies Optimization (CBO) is first introduced by Kaveh and Mahdavi [8] and already applied to many engineering problems [9]. According to CBO terminology, each candidate solution vector X_i is considered as a *Colliding Body*, CB. Such CB's consitiute the population of search agents which is subdivided into two groups; i.e. stationary and moving CB's. Every moving object moves toward and collide with the corresponding stationary body according to the CBO process. It is done for two purposes: (i) to improve the positions of moving objects and (ii) to push stationary objects towards better positions. Positions of colliding bodies are updated using their new velocities after the collision. Algorithmic steps of CBO can be briefed as follows:

1. *Initiation*: Randomly generate positions of n colliding bodies as the initial population by:

$$X_i = X^L + rand \circ \left(X^U - X^L\right) , \qquad i = 1, 2, \ldots, n \tag{6}$$

X^L and X^U denote lower and upper bounds on the design vector, repectively. *rand* is a random vector uniformly distributed in the range [0, 1] and the sign" $\circ$ " denotes the element-by-element multiplication. Mass of every *i*th colliding body (CB) is also defined as:

$$m_i = \frac{F(X_i)}{\sum_{i=1}^{n} F(X_i)} \qquad i = 1, 2, \ldots, n \tag{7}$$

where $F(X_i)$ represents fitness extracted out of the cost function $f(X_i)$, for the *i*th solution vector, X_i. It is assumed that a CB with good value has a larger mass than bad CB's.

2. Sort CB's in ascending order of their cost function values. Then subdivide the population into the following groups:

- *Stationary* CB's: the lower half of the sorted population. The velocity of every stationary CB before collision is initiated by zero:

$$V_i = 0 \quad , \quad i = 1, \ldots, \frac{n}{2} \tag{8}$$

- The upper half include *moving* CB's; i.e. those which move toward the stationary ones. Therefore, velocity of every moving CB before collision is given by:

$$V_i = X_i - X_{i-\frac{n}{2}} \quad , \quad i = \frac{n}{2} + 1, \ldots, n \tag{9}$$

in which $X_{i-\frac{n}{2}}$ represent the CB moving toward the corresponding *i*th stationary CB.

3. The velocity of every moving CB after the collision; V'_i is obtained by:

$$V_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) V_i}{m_i + m_{i-\frac{n}{2}}} \qquad , \; i = \frac{n}{2} + 1 \,, \ldots, n \tag{10}$$

where m_i and $m_{i-\frac{n}{2}}$ are masses of stationary and moving CB's, respectively. In addition, velocity of the corresponding stationary CB after the collision will be:

$$V_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) V_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} \qquad , \; i = 1 \,, \ldots, \frac{n}{2} \tag{11}$$

In order to provide a balance between exploration and exploitation the COR is linearly decreased from unity to zero. Thus, ε is given by:

$$\varepsilon = 1 - \frac{k}{N_I} \tag{12}$$

where k and N_I represent the current and total number of iterations, respectively.

4. New positions of CBs are evaluated using the velocities generated after the collision in position of stationary CBs. The new positions of each moving CB is:

$$x_i^{new} = x_{i-\frac{n}{2}} + rand \circ v_i' \quad , \quad i = \frac{n}{2} + 1, \ldots, n \tag{13}$$

where x_i^{new} and v_i' are the new position and the velocity after the collision of the ith moving CB, respectively; $X_{i-\frac{n}{2}}$ is the old position of ith stationary CB pair. Also, the new positions of stationary CBs are obtained by:

$$x_i^{new} = x_i + rand \circ v_i' \quad , \quad i = 1, \ldots, \frac{n}{2} \tag{14}$$

where x_i^{new}, x_i and v_i' are the new position, old position and the velocity after the collision of the ith stationary CB, respectively.

5. The optimization is repeated from Step 2 after mass update until the iteration number reaches N_I. It should be noted that, a body's status (stationary or moving body) and its numbering may be changed in any two subsequent iterations.

MATLAB codes for CBO and the enhanced version of it has already been given by Kaveh and Ilchi-Ghazaan [20].

3. OPPOSITION-BASED COLLIDING BODIES OPTIMIZATION

Several meta-heuristic algorithms initiate with a population of solution candidates and improve their best solution during the search until the optimum is captured. Such a process terminates as soon as some predefined criteria are satisfied. In the absence of a priori information about the solution, we usually start with random initial population. The computation time, among the others, is related to the distance of such an initial guess from the optimal solution. The closer this guess is to the global optimum, the quicker convergence of the algorithm and the higher quality of final solution is excpected.

One can improve the chance of starting with a closer (fitter) solution by simultaneously checking the opposite solution [15]. This way, the fitter one (an individual or its opposite) can be chosen for further progress. By almost equal likelihood it is expected that a search agent is farther from the true solution than its opposite and vice versa [21]. Therefore, taking into account the fittest of these opposite agents, will potentially accelerate the convergence. The approach can be applied not only to initial population but also to every solution arising during the search. The mathematical definition of OBL can be addressed as follows.

3.1 Opposition - based learning

Let $F\left(X\right)$ be a fitness function. It is supposed that solutions with higher fitness values are more desired. Let X be a primary guess and $\tilde{X}$ is its opposite vector, then in every iteration calculate $F\left(X\right)$ and $F\left(\tilde{X}\right)$. The learning continues with X if $F\left(X\right) \geq F\left(\tilde{X}\right)$, otherwise with $\tilde{X}$. Such a learning scheme is analogous to replacing the current vector with the fitter one of its position and the opposition vector.

3.2 Proposed opposition-based learning in the colliding bodies optimization

In the present work two types of opposition are utilized called *static opposition* and *dynamic opposition*. Static opposite of a CB position is obtained as its mirror picture with respect to the middle of the prescribed upper and lower bounds. It is mathematically defined by:

$$\tilde{X}^{S} = X^{L} + X^{U} - X^{CB} \tag{15}$$

where $\tilde{X}^{S}$ stands for the static opposite of X^{CB} as the colliding body.

In the dynamic opposition the location of mirror point is dynamically altered during optimization. In the present study, such a dynamic opposite of any *Moving CB* is obtained by mirroring its position with respect to the corresponding *Stationary CB*. That is:

$$\tilde{X}^{D} = 2X^{StationaryCB} - X^{MovingCB} \tag{16}$$

The aforementioned definitions are employed to develop *Opposition-based Colliding Bodeis Optimization,* OCBO via the following algorithmic steps:

1. Randomly initiate a population, Pop of n CB's using Eq. (6). Evaluate fitness and mass of all CB's in Pop.
2. Sort Pop in desceding order of the fitness values. Label the fitter half of Pop as $Pop_{Stationary}$ and the remainder as Pop_{Moving}.
3. For every CB vector; X_i in the $Pop_{Stationary}$ generate a newcomer solution; X_{i+n} by the following subroutine and add it to the current Pop to obtain $Pop_{auxiliary}$:

a. With the probability of one-fifth, calculate X_{n+i} from the corresponding X_i by dynamic opposition rule as in Eq. (16). That is:

$$X_{i+n} = 2X_i - X_{i+n/2} \tag{17}$$

b. Otherwise:

i. generate $\tilde{X}_i^S$ as the static opposite of X_i by Eq. (15)

ii. perform a crossover between X_i and $\tilde{X}_i^S$ and select X_{i+n} as the fittest one of the resulting children

4. Once $Pop_{auxiliary}$ is completed, sort it in desceding order of fitness values. Select the the first $\frac{n}{2}$ members of this sorted $Pop_{auxiliary}$ as the starionary CB's and the second $\frac{n}{2}$ vectors as the moving ones.
5. Update positions of moving and stationary CB's by Eq. (13) and Eq. (14), respectively.
6. Iterate the above procedure from step 3 until termination criterion is satisfied; that is the iteration number reaches N_I. Then announce the fittest obtained CB as the optimum solution X^* with the fitness: F^*.

Fig. 1, shows flowchart of the proposed OCBO algorithm. Note that at any iteration, $Pop_{auxiliary}$ is reduced to Pop with the fixed size of n. Using an elitist strategy the fittest CB is saved and updated via iterations of the search. It is worth mentioning that both OCBO and CBO have only two control parameters: the population size and the number of iterations. Other variants of CBO which have more control parameters are not considered in this study.

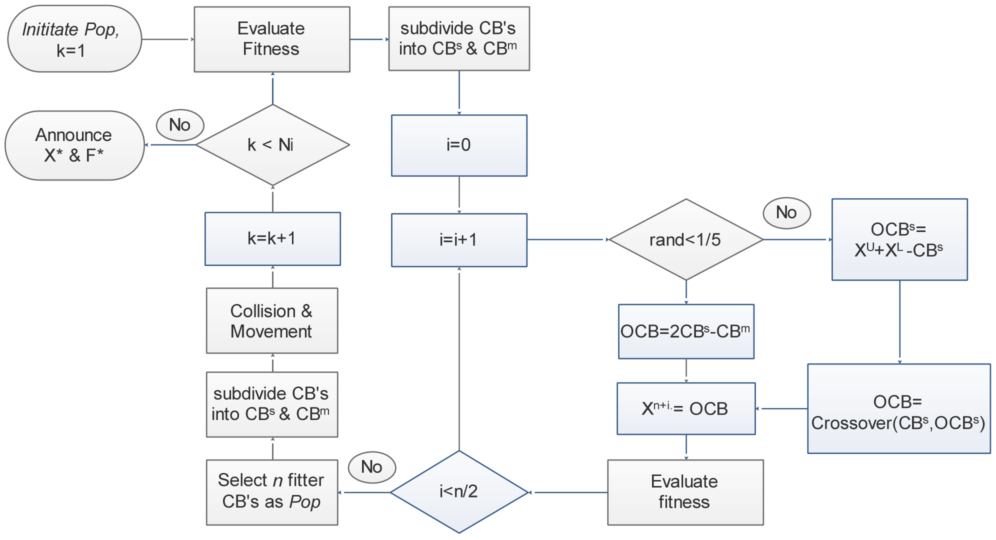


Figure 1. Flowchart of the proposed OCBO

4. UNCONSTRAINED OPTIMIZATION

In order to validate performance of OCBO, a number of test functions are selected from literature via four distinct classes [22, 23]: separable unimodal Table 1, non-separable unimodal Table 2, separable multimodal Table 3, and non-separable multimodal Table 4.

Table 1: Unimodal and separable test functions

Function ID	Name	Expression	Dim	Domain$\left[x_i^{LB}, x_i^{UB}\right]$	Global minimum
F1	Sphere	$F_1(\underline{X}) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	0
F2	Step	$F_2(\underline{X}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-100,100]^n$	0
F3	Quartic	$F_3(\underline{X}) = \sum_{i=1}^{n} i x_i^4 + random(0,1)$	30	$[-1.28,1.28]^n$	0

Table 2: Unimodal and non-separable test functions

Function ID	Name	Expression	Dim	Domain$\left[x_i^{LB}, x_i^{UB}\right]$	Global minimum
F4	Schwefel 2.22	$F_4(\underline{X}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10,10]^n$	0
F5	Schwefel 1.2	$F_5(\underline{X}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$[-100,100]^n$	0
F6	Schwefel 2.21	$F_6(\underline{X}) = \max \left\{ \lvert x_i \rvert 1 \le i \le n \right\}$	30	$[-100,100]^n$	0
F7	Rosenbrock	$F_7(\underline{X}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$	30	$[-30,30]^n$	0

Table 3: Multimodal and separable test functions

Function ID	Name	Expression	Dim	Domain $\left[x_i^{LB}, x_i^{UB}\right]$	Global minimum
F8	Schwefel	$F_8(\underline{X}) = \sum_{i=1}^{n} - x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500,500]^n$	-12569.5
F9	Rastrigin	$F_9(\underline{X}) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10 \right]$	30	$[-5.12,15.12]^n$	0
F10	Foxholes	$F_{10}(\underline{X}) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{j=1}^{2} \left(x_i - a_{ij}\right)^6} \right)^{-1}$	2	$[-65.53,65.53]^n$	1
F11	Branin	$F_{11}(\underline{X}) = \left(x_2 - \frac{5.1}{4\pi} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5,10] \times [0,15]$	0.398

Unimodal test functions have single optimum so they can benchmark the exploitation and convergence speed of the algorithm. Multimodal test functions have more than one optimum; the best is called global optimum while the rest are called local optima. An algorithm should properly balance exploration and exploitation to approximate the global optima. In non-separable functions each variable of a function is independent of the other variables. Separable functions are generally easier to solve than non-separable functions.

For all the test functions, a unified problem formulation is applied as:

$$\begin{aligned} &\textit{Maximize} \quad F(\underline{X}) = -f(\underline{X}) \\ &\textit{Subject to} \quad x_i^L \le x_i \le x_i^U, i = 1, \ldots, N \end{aligned} \tag{18}$$

whereas $\underline{X} = < x_i >$ is the vector of N design variables within the range $[x^L, x^U]$. The cost function and fitness function are denoted by $f(x)$ and $F(x)$, respectively.

Table 4: Multimodal and non-separable test functions

Function ID	Name	Expression	Dim	Domain $\left[x_i^{LB}, x_i^{UB}\right]$	Global minimum
F12	Ackley	$F_{12}(\underline{X}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32,32]^n$	0
F13	Griewank	$F_{13}(\underline{X}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600,600]^n$	0
F14	Penalized	$F_{14}(\underline{X}) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right] + (y_n-1)^2\right\}$ $+\sum_{i=1}^{n} u(x_i,10,100.4)$, $y_i = 1 + \frac{x_i+1}{4}$, $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & , x_i>a \\ 0 & , -a<x_i<a \\ k(-x_i-a)^m & , x_i<-a \end{cases}$	30	$[-50,50]^n$	0
F15	Penalized 2	$F_{15}(\underline{X}) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_1+1)\right] + (x_n-1)^2\left[1+\sin^2(2\pi x_2)\right]\right\}$ $+\sum_{i=1}^{n} u(x_i,5,100,4)$, $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & , x_i>a \\ 0 & , -a<x_i<a \\ k(-x_i-a)^m & , x_i<-a \end{cases}$	30	$[-50,50]^n$	0
F16	Kowalik	$F_{16}(\underline{X}) = \sum_{i=1}^{11}\left[a_i \frac{x_i(b_i^2+b_i x_2}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5,5]^n$	0.00031
F17	6- Hump Camel Back	$F_{17}(\underline{X}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5,5]^n$	0.03162
F18	GoldStein-Price	$F_{18}(\underline{X}) = \left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]$ $\times\left[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$	2	$[-2,2]^n$	3
F19	Hartman 3	$F_{19}(\underline{X}) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j-p_{ij})^2\right)$	4	$[0,1]^n$	-3.86
F20	Hartman 6	$F_{20}(\underline{X}) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j-p_{ij})^2\right)$	6	$[0,1]^n$	-3.32
F21	Shekel 5	$F_{21}(\underline{X}) = -\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^t + c_i\right]^{-1}$	4	$[0,10]^n$	-10
F22	Shekel 7	$F_{22}(\underline{X}) = -\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^t + c_i\right]^{-1}$	4	$[0,10]^n$	-10
F23	Shekel 10	$F_{23}(\underline{X}) = -\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^t + c_i\right]^{-1}$	4	$[0,10]^n$	-10

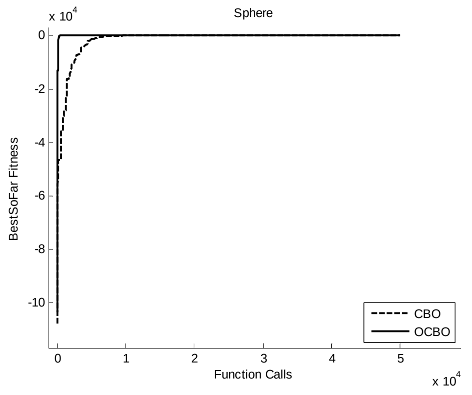
After a few trials the control paramters are set to $n = 50$, $N_I = 1000$. Every such function is optimized within 50 independent runs provided that the initial population is kept identical for different algorithms in each run. Consequently, statistical results are derived in terms of the best, mean and standard deviation.

Table 5: Optimal fitness results for unimodal and separable test functions

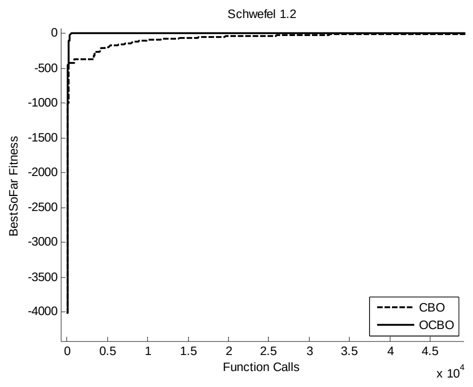
Function ID	Statistical Item	CBO	OCBO
F1	Best	-1.51002e-13	0
	Mean	-1.68668e-10	0
	Standard deviation	1.00227e-09	0
F2	Best	0	0
	Mean	-0.1	0
	Standard deviation	0.30305	0
F3	Best	-0.01005	-3.4543e-07
	Mean	-0.03688	-2.6266e-05
	Standard deviation	0.01438	2.4392e-05

Table 6: Optimal fitness results for unimodal and non-separable test functions

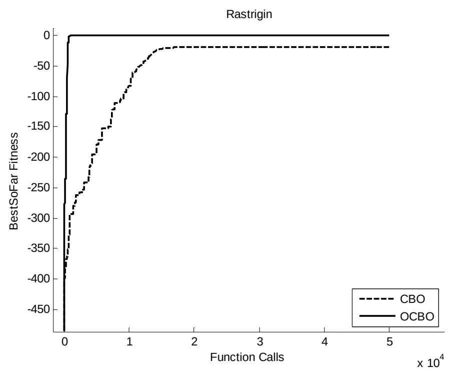
Function ID	Statistical Item	CBO	OCBO
F4	Best	-3.13092e-10	0
	Mean	-1.48948e-09	0
	Standard deviation	1.64468e-09	0
F5	Best	-1.31271	0
	Mean	-4.02597	0
	Standard deviation	1.94434	0
F6	Best	-12.39720	0
	Mean	-35.88630	0
	Standard deviation	10.29710	0
F7	Best	-10.70481	-25.84623
	Mean	-118.17462	-26.22199
	Standard deviation	201.26144	0.14224



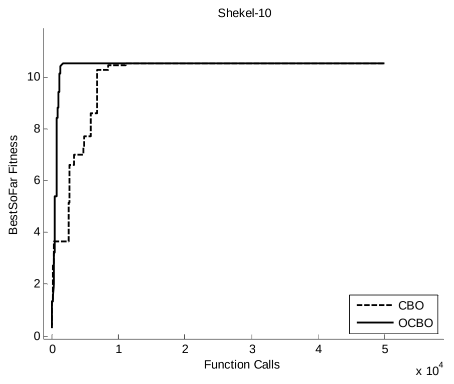
F1



F5



F9



F23

Figure 2. Sample convergence trace of F1, F5, F9 and F23 in various classes of test functions

According to the results for unimodal separable test functions in Table 5, it is evident that OCBO has outperformed CBO regarding quality of final solutions. Note that the cost values less than 10^{-17} are rounded to 0. Table 6 compares the statistical results of the present methods for unimodal and non-separable test functions. For F4, F5 and F6, OCBO has been superior in terms of the best and mean results. Besides, its standard deviation has been less than CBO. Although the best result of CBO has been better in F7, its mean and standard deviation is outperformed by OCBO.

Such a comparison is repeated for multimodal separable functions. According to Table 7, OCBO has been the winner for F8, F9 and F10. However for F11, both the present algorithms have revealed similar results.

The results of 50 runs for multimodal and non-separable test functions are given in Table 8, It can be noticed that except for F18 and F20, OCBO has outperformed CBO. For F18 both have similar results. Treating F20 function, the mean cost and standard deviation of CBO has been better, however, both has obtained the best result. Fig.2 exhibits sample convergence trace of the optimization methods in each class of the treated test functions.

Table 7: Optimal fitness results for multimodal and separable test functions

Function ID	Statistical Item	CBO	OCBO
F8	Best	5615.77990	5417.67480
	Mean	5369.45720	4227.63440
	Standard deviation	216.06371	312.87596
F9	Best	-18.90422	0
	Mean	-37.88798	0
	Standard deviation	11.16805	0
F10	Best	-0.99800	-0.99800
	Mean	-3.33772	-3.23043
	Standard deviation	2.61819	2.07934
F11	Best	-0.397889	-0.39789
	Mean	-0.39789	-0.39789
	Standard deviation	3.36448e-16	3.36448e-16

Table 8: Optimal fitness results for multimodal and non-separable test functions

Function ID	Statistical Item	CBO	OCBO
F12	Best	-8.53276e-08	-8.88178e-16
	Mean	-0.17404	-8.88178e-16
	Standard deviation	0.44397	0
F13	Best	-4.77396e-13	0
	Mean	-0.00752	0
	Standard deviation	0.01113	0
F14	Best	-4.75800e-15	-1.12272e-13
	Mean	-0.06012	-1.67474e-12
	Standard deviation	0.10900	2.26344e-12
F15	Best	-1.07110e-14	-5.23943e-20
	Mean	-0.04876	-0.03183
	Standard deviation	0.18447	0.17367
F16	Best	-0.00068	-0.00032
	Mean	-0.00129	-0.00061
	Standard deviation	0.00182	0.00012
F17	Best	1.031628	1.031628
	Mean	1.031627	1.031628
	Standard deviation	6.41006e-06	2.24299e-16
F18	Best	-3	-3

	Mean	-3	-3
	Standard deviation	4.28781e-15	4.35625e-15
F19	Best	3.86278	3.86278
	Mean	3.86278	3.86278
	Standard deviation	3.14018e-15	3.14018e-15
F20	Best	3.32237	3.32237
	Mean	3.32237	3.27325
	Standard deviation	1.34579e-15	0.05868
F21	Best	10.15320	10.15320
	Mean	5.16850	9.55795
	Standard deviation	2.60023	1.24359
F22	Best	10.40290	10.40290
	Mean	9.88457	9.97743
	Standard deviation	1.80663	1.35162
F23	Best	10.53640	10.53640
	Mean	9.94920	10.22230
	Standard deviation	2.01901	1.27639

5. CONSTRAINED STRUCTURAL OPTIMIZATION

Structural optimization is usually addressed with narrow feasible regions and complex function analyses. Thus, optimal design of truss and frame structures is treated to validate capability of the proposed OCBO in solving constrained problems.

As a common practice, structural weight minimization is formulated subject to the stress and displacement constraints. It is utilized here via the following penalized function:

$$Maximize\ F(\underline{X}) = -W*(1+Kp*\sum_{i=1}^{m} C_i) \tag{19}$$

W denotes the total structural weight, *Kp,* is the penalty coefficient desired by the user and C_i stands for the violation of the i^{th} stress or displacement constraint. The design vector $\underline{X}$ includes section indices (or areas) for sizing design, however, for geometry part of optimization, nodal coordinates are also utilized as design variables.

The same control parameters are applied and the maximum number of function evaluations is set to 10000. Statistical results of truss examples are derived from 50 independent runs with $Kp = 10$.

Table 9: Available sections for the 52-bar truss

No.	$10^{-6}m^2$	No.	$10^{-6}m^2$	No.	$10^{-6}m^2$	No.	$10^{-6}m^2$
1	71.613	17	1008.385	33	2477.414	49	7419.340
2	90.968	18	1045.159	34	2496.769	50	8709.660
3	126.451	19	1161.288	35	2503.221	51	8967.724
4	161.290	20	1283.868	36	2696.769	52	9161.272
5	198.064	21	1374.191	37	2722.575	53	9999.980
6	252.258	22	1535.481	38	2896.768	54	10322.560
7	285.161	23	1690.319	39	2961.284	55	10903.204
8	363.225	24	1696.771	40	3096.768	56	12129.008
9	388.386	25	1858.061	41	3206.445	57	12838.684
10	494.193	26	1890.319	42	3303.219	58	14193.520
11	506.451	27	1993.544	43	3703.218	59	14774.164
12	641.289	28	729.031	44	4658.055	60	15806.420
13	645.160	29	2180.641	45	5141.925	61	17096.740
14	792.256	30	2238.705	46	5503.215	62	18064.480
15	816.773	31	2290.318	47	5999.988	63	19354.800
16	939.998	32	2341.931	48	6999.986	64	21612.860

5.1 Sizing design of the 52-bar truss

As the first engineering example the planar truss of Fig.3, is optimized for minimal weight. The structural members are divided into 12 groups: (1) $A_1 - A_4$, (2) $A_5 - A_{10}$, (3) $A_{11} - A_{13}$, (4) $A_{14} - A_{17}$, (5) $A_{18} - A_{23}$, (6) $A_{24} - A_{26}$, (7) $A_{27} - A_{30}$, (8) $A_{31} - A_{36}$, (9) $A_{37} - A_{39}$, (10) $A_{40} - A_{43}$, (11) $A_{44} - A_{49}$, (12) $A_{50} - A_{52}$. Material density of steel is taken 7860 kg/m^3 and modulus of elasticity is 2.07×10^5 Mpa. The member stresses are limited to ± 180 Mpa in both tension and compression. Static loading of $P_x = 100\,kN$ and $P_y = 200\,kN$ is exerted on the nodes 17, 18, 19 and 20.

The discrete variables are selected from Table 9. This benchmark problem has already been treated by Li et al. [24] using HPSO, Sadollah et al. [25] using MBA, Lee et al. [26] using HS, Cheng and Prayogo [27] using SOS and Wu and Chaw [28] using GA. The best results among them are given in Table 10; next to those obtained by the present works.

The first rank in this optimization belongs to OCBO which has obtained optimal weight of 1901.567kg by just 6525 structural analyses. NFEL indicates the required number of fitness evaluations up to the last improvement during optimization. CBO stands at the second rank with 1902.606kg by 8000 function calls while the other literature works resulted in higher weights even with 100000 analyses Table 10.

Further comparsion of the present methods confirms superiority of OCBO both in the worst run and mean result. Moreover, it is more robust due to its lower standard deviation of 91.5kg with respect to 187.0kg by CBO.

In order to better study trend of balancing between exploration and exploitation, sum of the velocity norms for CB's at each generation is calculated and traced via iterations of the search. In this regard, Fig.4 shows higher diversity at early stages of OCBO followed by more rapid convergence than CBO, to zero velocity of CB's as the search progress.

Table 10: Optimization results for the 52-bar truss

Element group	Wu and Chow, GA [28]	Li et al. PSO [24]	Cheng and Prayogo, SOS [27]	Present work	
				CBO	OCBO
G1	4658.055	4658.055	4658.055	4658.055	4658.055
G2	1161.288	1161.288	1161.288	1161.288	1161.288
G3	645.16	363.225	494.193	494.193	506.451
G4	3303.219	3303.219	3303.219	3303.219	3303.219
G5	1045.159	940.000	940.000	939.998	939.998
G6	494.193	949.193	494.193	494.193	506.451
G7	2477.414	2238.705	2238.705	2238.705	2238.705
G8	1045.159	1008.385	1008.385	1008.385	1008.385
G9	285.161	388.386	494.193	494.193	388.386
G10	1696.771	1283.868	1283.868	1283.868	1283.868
G11	1045.159	1161.288	1161.288	1161.288	1161.288
G12	641.289	792.256	494.193	494.193	506.451
Best weight (kg)	1970.142	1905.49	1902.605	1902.6055	1901.5671
Mean weight (kg)	-	-	-	2087.2509	1943.3098
Worst weight (kg)	-	-	-	2645.216	2156.5225
Standard deviation	-	-	-	186.9582	91.5630
NFEL	60,000	100,000	-	8000	6525

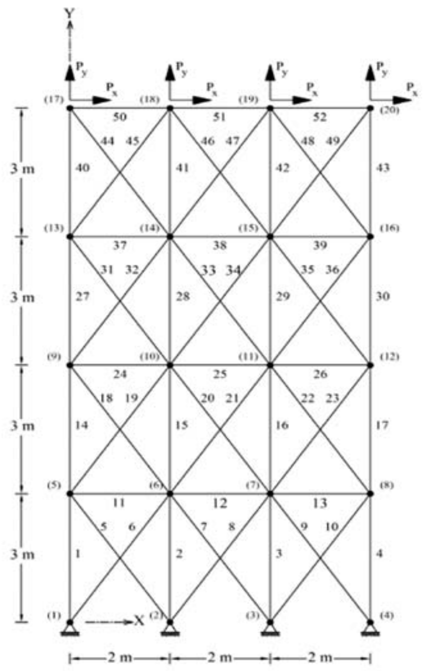


Figure 3. The 52-bar truss [25]

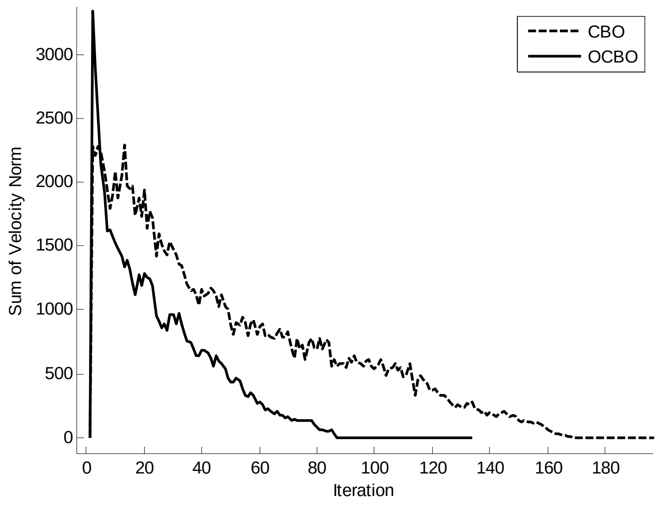


Figure 4. Diversity trace for the 52-bar truss example

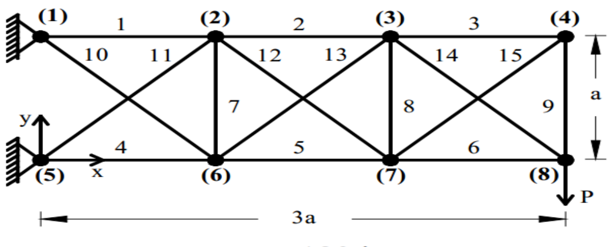


Figure 5. The 15-bar truss [30]

5.2 Geometry and sizing design of the 15-bar truss

Both member sizing and nodal geometry for the 15-bar truss of Fig.5 is optimized in this example. Material density is $0.1\ lb/in^3\ (2720\,kg/m^3)$ and the modulus of elasticity is $10000\,\text{ksi}\ (68.947\,\text{GPa})$. The stress limit for every truss member is $25\,\text{ksi}$ in tension and compression. Both x and y coordinate of the nodes 2, 3, 6 and 7, are taken geometry design variables. The nodes 6 and 7 have the same x coordinates as the nodes 2 and 3, respectively. For the nodes 4 and 8, just y-coordinates are considered as design variables. Thus, the design vector constitutes 15 sizing variables (cross-sectional area of bars) and 8 geometry variables ($x_2{=}x_6, x_3{=}x_7, y_2, y_3, y_4, y_6, y_7, y_8$). The availabe profile list for sizing is given by S = {0.111, 0.141, 0.174, 0.22, 0.27, 0.287, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.656, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18} in^2. Table 11 reveals geometry variable bounds.

Table 11: Bounds of geometric variables for the 15-bar truss design

Design variable	Lower bound (in)	Upper bound (in)
X2	100	140
X3	220	260
Y2	100	140
Y3	100	140
Y4	50	90
Y6	-20	20
Y7	-20	20
Y8	20	60

Table 12: Optimization results for the 15-bar truss

Design variable	Wu and Chow, GA [28]	Tang et al. GA [33]	Hwang et al. ARSAGA [31]	Present work CBO	Present work OCBO
Sizing variables (in^2)					
A1	1.174	1.081	0.954	1.081	0.954
A2	0.954	0.539	1.081	0.954	0.954
A3	0.440	0.287	0.440	0.141	0.270
A4	1.333	0.954	1.174	1.174	1.081
A5	0.954	0.954	1.488	0.954	0.539
A6	0.174	0.220	0.270	0.539	0.270
A7	0.440	0.111	0.270	0.111	0.111
A8	0.440	0.111	0.347	0.111	0.111
A9	1.081	0.287	0.220	0.111	0.287
A10	1.333	0.220	0.440	0.220	0.440
A11	0.174	0.440	0.220	0.174	0.287
A12	0.174	0.440	0.440	0.174	0.111
A13	0.347	0.111	0.347	0.270	0.270
A14	0.347	0.220	0.270	0.539	0.270
A15	0.440	0.347	0.220	0.141	0.270
Geometry variables (in)					
X2	123.189	133.612	118.346	120.9269	118.8259
X3	231.595	234.752	225.209	220.3005	239.0584
Y2	107.189	100.449	119.046	113.2465	128.9736
Y3	119.175	104.738	105.086	101.6757	112.2486
Y4	60.462	73.762	63.375	59.0421	50.9543
Y6	-16.728	-10.067	-20.000	14.4389	-3.2506
Y7	15.565	-1.339	-20.000	15.4493	8.3002
Y8	36.645	50.402	57.722	59.2315	51.2586
Best weight (lb)	120.528	79.820	104.573	80.7652	78.1435
Mean weight (lb)	-	-	-	84.0269	80.3756
Worst weight (lb)	-	-	-	89.1467	81.6599
Standard deviation	-	-	-	2.2316	0.7243
NFEL	-	-	-	9650	9900

This problem has been addressed by several investigators including Kamyab-Moghadas and Gholizadeh using CAFA [29], Kazemzade-Azad et al. using MBRCGA [30], Hewang and He using ARSAGA [31], Wu and Chow [32] and Tang et al. using GA [33].

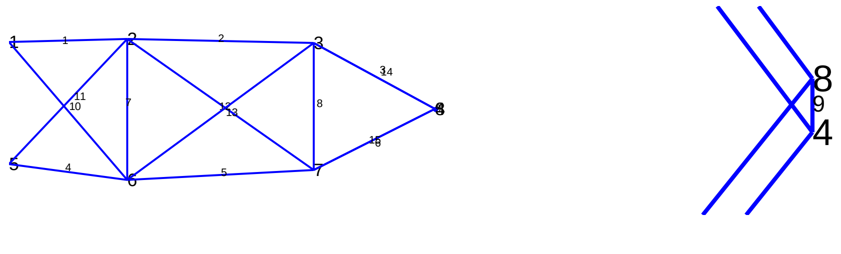


Figure 6. (a) Optimum geometry of the 15-bar truss by OCBO, (b) Position of the nodes 4, 8

The best optimal geometry has been achieved by OCBO as depicted in Fig.6. Acording to Table 12, in this example OCBO has captured the least weight of 78.14*lb* via 9900 NFEL while the best result of the other algorithms, is 79.80*lb* by GA [33]. The optimal weight by CBO; which is 80.76*lb* has not shown further improvement after NFEL of 9650 up to 10000 structural analyses. Table 12 also confirms superiority of OCBO in the worst, mean and standard deviation of the results. Fig.7, shows convergence trends toward zero velocity by OCBO and CBO.

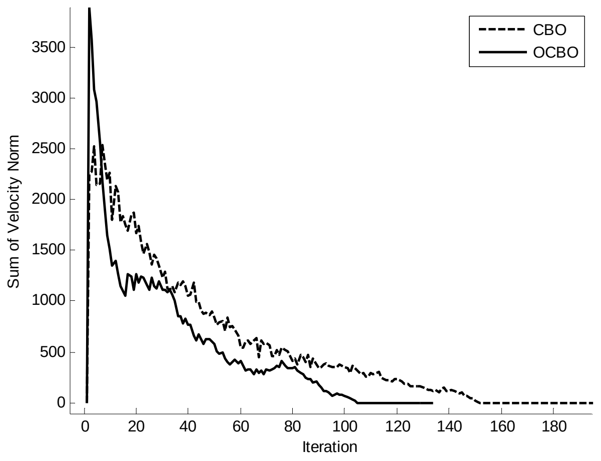


Figure 7. Diversity curves for the 15-bar truss example

5.3 Sizing optimization of the 3-bay 12-story frame via spectral design

In this example, the structure type is changed to the steel frame while both static gravitational and spectral seismic loading are taken into account by more rigorous analyses. Sizing design for the 3-bay 12-story ordinary moment frame is performed using available sections of Tables 13 and 14 for beams and columns, respectively. The story height and bay length are taken 3.3m and 5m, respectively. Frame consists of 84 members collected via 8 column groups and 4 beam groups as demonstrated in Fig. 8.

Table 13: Allowable sections for frame beams

No.	Profile	No.	Profile
1	IPB 10	13	IPB 34
2	IPB 12	14	IPB 36
3	IPB 14	15	IPB 40
4	IPB 16	16	IPB 45
5	IPB 18	17	IPB 50
6	IPB 20	18	IPB 55
7	IPB 22	19	IPB 60
8	IPB 24	20	IPB 65
9	IPB 26	21	IPB 70
10	IPB 28	22	IPB 80
11	IPB 30	23	IPB 90
12	IPB 32	24	IPB 100

For the seismic excitation, design spectrum of the Iranian seismic design code [34] is applied with the parameters: $A=0.35$, $Ru=5$, $I=1$ and soil type-II. Story masses are exerted on beams by a uniformly distributed load of 3500kgf/m. Feasibility of candidate solution, is checked due to LRFD design requirements [35].

Table 14: Allowable sections for frame columns

No.	Profile
1	Box30*30*1
2	Box30*30*2
3	Box30*30*3
4	Box35*35*1
5	Box35*35*2
6	Box35*35*3
7	Box40*40*1
8	Box40*40*2
9	Box40*40*3
10	Box45*45*1
11	Box45*45*2
12	Box45*45*3
13	Box50*50*1
14	Box50*50*2
15	Box50*50*3

According to Table 15, in this example OCBO has captured the mean result of 38326kg with standard deviation of 688kg; while CBO has resulted in an average of 38658kg and deviation of 1035kg. In addition, higher quality of the best solution for OCBO is revealed by Table 15 as is clearly observed in Fig.10a. For the sake of fair comparison, fitness values are depicted vs the number of function calls.

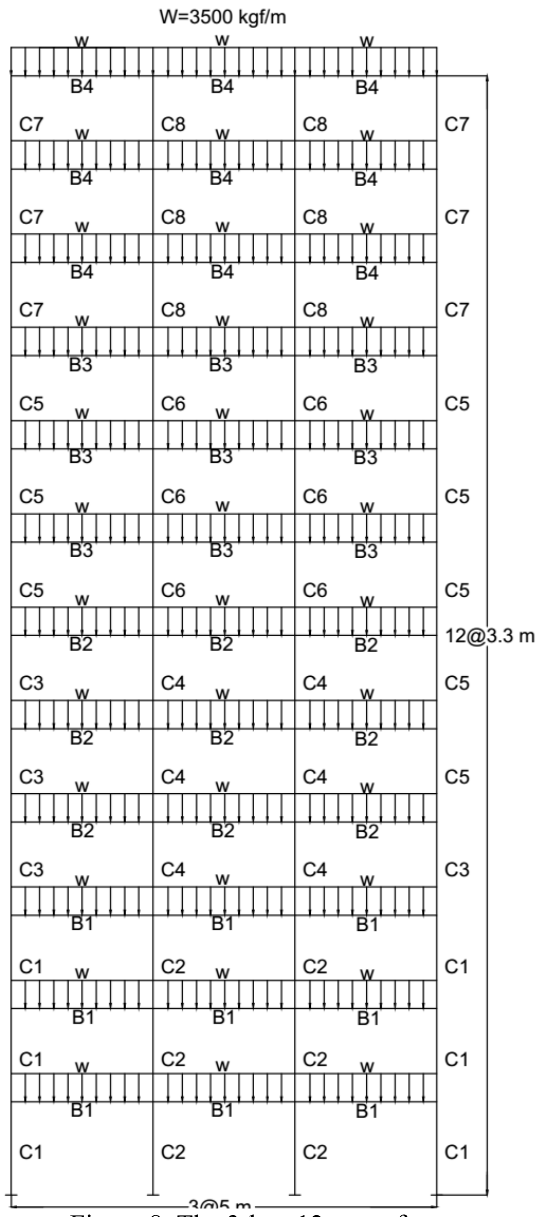
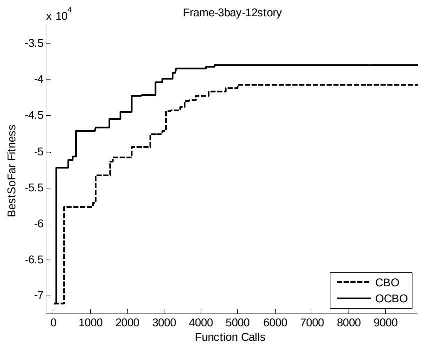


Figure 8. The 3-bay 12-story frame

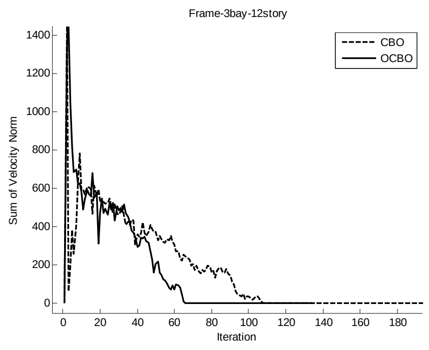
Fig. 9 shows that in this example, OCBO has not only higher convergence rate than CBO but also it has successfully overpassed local optima. The matter is related to the difference between the methods in trend of diversity decrease with iterations. As shown in Fig.9b, OCBO provides higher diversity in early iterations but more rapid convergence in the final ones, with respect to CBO. Such results address the effect of opposition-based learning in accelerated exploration of global optimum.

Table 15: Optimization results for the 3-bay 12-story frame

Element group	CBO	OCBO
C1	Box45*45*1	Box45*45*1
C2	Box45*45*1	Box45*45*1
C3	Box35*35*1	Box45*45*1
C4	Box45*45*1	Box40*40*1
C5	Box35*35*1	Box35*35*1
C6	Box35*35*1	Box35*35*1
C7	Box30*30*1	Box30*30*1
C8	Box30*30*1	Box30*30*1
B1	IPB 34	IPB 32
B2	IPB 30	IPB 30
B3	IPB 28	IPB 28
B4	IPB 22	IPB 22
Best weight (kg)	37301.7240	37297.5840
Mean weight (kg)	38657.5164	38325.7908
Worst weight (kg)	40706.3880	39737.0880
Standard deviation	1035.3129	687.8315



(a)



(b)

Figure 9. (a) Convergence and (b) diversity curves for the 3-bay 12-story frame

5.4 Earthquake clustering problem

An example of earthquake clustering is treated here as a discrete problem. Given a number of seismic records, the problem is to subdivide them in a prescribed number of groups; namely K clusters. Each entity (Earthquake record) is desired to be similar to the other entities in the same cluster but far from the ones in the other clusters. Similarity is measured by a distance metric over entity vectors. A common metric to evaluate clustering is:

$$s_i = \frac{q_i - p_i}{\max(q_i, p_i)} \tag{20}$$

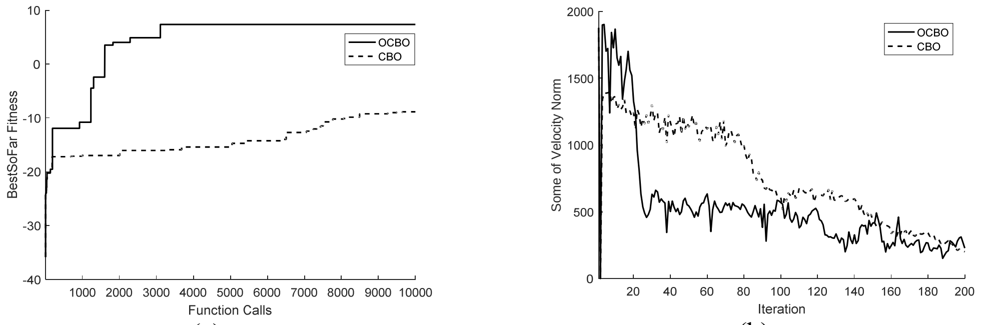
s_i stands for *silhouette value* of the i^{th} enitity; for which p_i and q_i denote the mean minimal with-in-cluster and maximal out-of-cluster distances; respectively. The silhouette value of each entity varies between -1 (the worst case) and +1 (the best case).

Consequently, the clustering problem is formulated here as follows:

$$Maximize\ f(\underline{X}) = \sum_{i=1}^{Ne} s_i \tag{21}$$

Every component x_i in the design vector; is the cluster number associated with the corresponding entity. It is an integer number limited between 1 and *K*. The optimum is the clustering result with maximum sum of silhouette values.

In this example, a datamatrix of $N_e = 100$ earthquake records on different soil types and magnitutes are selected from worldwide catalogue of PEER [36]. Euclidean distance is used in calculation of silhouette values; that is norm of difference vector between every two attribute vectors. A number of attributes are provided in the vector for each seismic record; including peak ground displacement, velocity and acceleration, effective duration, energy of the record, earthquake magnitude and soil-type of the site. The optimal clustering is searched by CBO and OCBO for $K = 10$. Result of clustering optimization is illustrated in Fig. 10. As can be realized, embedding OBL to CBO has considerably accelerated its convergence toward high quality results; even in such a discrete problem. The some of velocity norm in OCBO rapidly decreases after some early iterations; showing proper convergence trend.



(a) (b)

Figure 10. (a) Convergence and (b) diversity curves for Earthquake clustering problem

6. CONCLUSION

In the present work, opposition-based learning was applied at a parameter-less variant of colliding bodies optimization. Special static and dynamic types of OBL are offered. Additional exploitation is provided by embedding crossover to the static phase of learning. A fitness-based screening strategy is also employed to maintain fixed size of the population.

The proposed OCBO was found quite effective in capturing global optima via four distinct categories of unconstrained functions. In majority of the tests, the present method of

OBL has enhaced performance of CBO not only in the best but also for the mean results. The standrard deviation about the mean has also been lowered; exhipiting stable convergence of OCBO.

Performance test was repeated for more complicated constrained structural optimization. Examples of size and geometry optimization of trusses were treated under static loading with different constraints. Consequently, OCBO effectiveness was validated in capturing higher quality optima than the other literature works.

Defining a diversity measure, behavior of the treated algorithms were better studied; that is by sum of the population velocity norms vs iteration number. When such an index converges to zero, all the CB's are stable at their final positions and no further fitness improvement is observed. Tracing some of velocity norm index, reveals trend of exploration and exploitation of the algorithm in hand. Consequently, it was declared that OCBO exerts more exploration at early stages of the search as a result of embedding OBL to the CBO. Once the global optimum region was detected, OCBO rapidly converges toward it.

Treating an example of frame sizing by spectral dynamic analyses, it was shown that the embedded opposition-based learning has enabled the original method to overpass local optima and achive a considerably higher fitness level. Applying OBL has also enhanced the convergence rate of CBO in earthquakes clustering as a discerete practical problem.

The proposed method of OBL is found quite efficient to improve performance of the parameter-less CBO; however, other powerful variants of CBO are not considered here because of having more parameters. According to the numerical simulations, OCBO exhibits successive effectiveness in capturing high quality optima. In addition, it shows competitive mean results and convergence trend. Hence, the proposed method can be recommended for practice due to its few control parameters and efficiency in treated problems of unconstrained, constrained, continuous and discrete types.

Acknowledgement: The first author is grateful to Kharazmi University for the support (Ref.No.4.193535).

REFERENCES

1. Kaveh A. *Advances in Meta-heuristic Algorithms for Optimal Design of Structures*, Springer International Publishing, Switzerland, 2017.
2. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search, *Simulat* 2001; **76**(2): 60-8.
3. Manjarres D, Landa-Torres I, Gil-Lopez S, Del Ser J, Bilbao MN, Salcedo-Sanz S, Geem ZW. A Survey on Applications of the Harmony Search Algorithm, *Eng Appl Artif Int* 2013; **26**(8): 1818-31.
4. Shahrouzi, M. Optimal Spectral Matching of Strong Ground Motion by Opposition-Switching Search, In: Rodrigues, H., et. al. (eds), *Proc. 6th International Conference on Engineering Optimization (EngOpt 2018), Springer*, Cham, 2019; 713-724.
5. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**: 267-89.
6. Shahrouzi M. Pseudo-random directional search: a new heuristic for optimization, *Int J Optim Civil Eng* 2011; **1**(2): 341-355.

7. Kaveh A, Zolghadr A. A novel meta-heuristic algorithm: Tug of war optimization, *Int J Optim Civil Eng* 2016; **6**(4): 469-92.
8. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method, *Comput Struct* 2014; **139**: 18-27.
9. Kaveh A, Mahdavi VR. *Colliding Bodies Optimization: Extensions and Applications*, Springer International Publishing, Switzerland, 2015.
10. Kaveh A, Bakhshpoori T. Water evaporation optimization: A novel physically inspired optimization algorithm, Comput Struct 2016; **167**: 69-85.
11. Shahrouzi M, Aghabaglou M, Rafiee F. Observer-teacher-learner-based optimization : An enhanced meta-heuristic for structural sizing design, *Struct Eng Mech* 2017; **62**: 537-50.
12. Shahrouzi M, Pashaei M. Stochastic directional search: An efficient heuristic for structural optimization of building frames, *Sci Iran* 2013; **20**(4): 1124-32.
13. Kaveh A, Ilchi-Ghazaan M. A new meta-heuristic algorithm: vibrating particles system, *Sci Iran* 2017; **24**: 1-32.
14. Kaveh A. *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, Springer International Publishing; Switzerland, 2017.
15. Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence, *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Austria, 2005; pp. 695-701.
16. Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition-based differential evolution, *IEEE Trans Evol Comput* 2008; **12**(1): 64-79.
17. Singh RP, Mukherjee V, Ghoshal SP. The opposition-based harmony search algorithm, J. *Inst Eng India Ser: B* 2014; **94**(4): 247-56.
18. Shaw B, Mukherjee V, Ghoshal SP. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, *Int J Electr Power Ener Syst* 2012; **35**(1): 21-33.
19. Rahnamayan S, Wang GG. Solving large scale optimization problems by opposition-based differential evolution (ODE), *WSEAS Trans Comput* 2008; **7**: 1792-1804.
20. Kaveh A, Ilchi-Ghazaan M. Computer codes for colliding bodies optimization and its enhanced version, *Int J Optim Civil Eng* 2014; **4**(3): 321-39.
21. Tizhoosh HR. Reinforcement learning based on actions and opposite actions, *ICGST International Conference Artificial Intelligence and Machine Learning*, Cario, Egypt, 2005.
22. Yao X, Liu Y, Lin G. Evolutionary programming made faster, *IEEE Trans Evol Comput* 1999; **3**: 82-102.
23. Jamil M, Yang X. A literature survey of benchmark functions for global optimization problems, *Int J Math Model Num Optim* 2013; **4**(2): 150-94.
24. Li LJ, Huang ZB, Liu FA. Heuristic particle swarm optimization method for truss structures with discrete variables, *Comput Struct* 2009; **87**: 435-43.
25. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm for optimization of truss structures with discrete variables, *Comput Struct* 2012; **102**: 49-63.
26. Lee KS, Geem ZW, Lee SH, Bae KW. The harmony search heuristic algorithm for discrete structural optimization, *Eng Optim* 2005; **37**: 663-84.

27. Cheng MY, Prayogo D. Symbiotic organism search: A new metaheuristic optimization, *Comput Struct* 2014; **139**: 98-112.
28. Wu SJ, Chow PT. Steady-state genetic algorithm for discrete optimization of trusses, *Comput Struct* 1995; **56**: 979-91.
29. Kamyab-Moghadas R, Gholizadeh S. A cellular automata firefly algorithm for layout optimization of truss structures, *Int J Optim Civil Eng* 2017; **7**: 13-23.
30. Kazemzadeh-Azad S, Kazemzadeh-Azad S, Jayant-Kulkarni A. Structural optimization using a mutation-based genetic algorithm, *Int J Optim Civil Eng* 2012; **2**(1): 80-100.
31. Hwang SF, He RS. A hybrid real-parameter genetic algorithm for function optimization, *Adv Eng Info* 2006; **20**: 7-21.
32. Wu SJ, Chow PT. Integrated discrete and configuration optimization of trusses using genetic algorithm, *Comput Struct* 1995; **55**(4): 695-702.
33. Tang W, Tong L, Gu Y. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables, *Int Numer Mest Eng* 1995; **62**: 1737–62.
34. Building and Housing Research Center, *Iranian Code for Seismic Resistant Design of Building: Standard-2800*, 4th Ed, Tehran, Iran, 2016.
35. INBC: Part-10, *Iranian National Building Code, Part-10: Design of Steel Structures*, 4th ed. Roads, Housing and Urban Development of Iran, Tehran, Iran, 2013.
36. Pacific Earthquake Engineering Research center, PEER ground motion database, https://ngawest2.berkeley.edu.