ASIAN JOURNAL OF CIVIL ENGINEERING (BHRC) VOL. 18, NO. 4(2017)
PAGES 657-671

OPTIMAL DESIGN OF REINFORCED CONCRETE CANTILEVER RETAINING WALLS USING CBO, ECBO AND VPS ALGORITHMS

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

Received: 29 September 2016; **Accepted:** 10 December 2016

ABSTRACT

In this study optimal design of reinforced concrete cantilever retaining walls is performed under static and earthquake loading conditions utilizing the Colliding Bodies of Optimization (CBO), Enhanced Colliding Bodies of Optimization (ECBO) and vibrating particles system (VPS) methods. This design is based on ACI 318-05 and two theories known as Coulomb and Rankine have been applied for estimating the earth pressures under static loading condition, and Mononobe-Okabe method have been applied for estimating earth pressures under earthquake loading condition. The objective function considered is the cost of the retaining wall and this function is minimized subjected to design constraints. The performances of the CBO, ECBO and VPS and some other optimization algorithms are compared for the considered benchmark examples.

Keywords: Reinforced concrete; cantilever retaining wall; colliding bodies optimization; enhanced colliding bodies optimization; vibrating particles system optimization.

1. INTRODUCTION

Optimization is one of the important issues in science of engineering and human have been using concepts of optimization for dealing with his/her problems. Optimization algorithms are utilized to minimize or maximize an objective function under certain specific limitations.

Retaining walls are structures to prevent the soil to fall or slide. These structures have wide range of applications and most commonly are used in construction of roads, bridges, transportation systems and other constructional facilities. In optimal design of retaining walls, the main objective is to minimize the cost or weight of the retaining walls under some constraints, such as stresses and displacements. Cantilever retaining walls are made of reinforced concrete consisting of a thin stem and a base slab. This type of wall is economical

*E-mail address of the corresponding author: alikaveh@iust.ac.ir (A. Kaveh)

up to a height of 8 m (25 ft) [1]. Some studies have been carried out in this field by Rhomberg et al. [2], Saribas and Erbatur [3], Ceranic et al. [4], Castillo et al. [5], Basudhar et al. [6], Sivakumar [7], Yepes et al. [8], Camp and Akin [9]. Kaveh and Shakouri [10] used harmony search based algorithm for the optimum cost design of reinforced concrete cantilever retaining walls and Kaveh and Behnam [11] utilized Charged System Search algorithm for optimization of cantilever retaining walls. Kalateh et al. [12] utilized multi-objective genetic algorithm for design of a retaining wall under seismic loads. Kaveh and Khayatazad [13] employed ray optimization method for optimal design of retaining walls. Kaveh and Soleimani [14] utilized CBO and DPSO algorithms for optimal design of cantilever retaining walls. Kaveh and Farhoudi [15] employed dolphin echolocation optimization for design of cantilever retaining walls.

In this study the colliding bodies optimization (CBO) developed by Kaveh and Mahdavi [16], and the enhanced colliding bodies optimization (ECBO) introduced by Kaveh and Ilchi Ghazaan [17], and vibrating particles system (VPS) developed by Kaveh and Ilchi Ghazaan [18] are utilized to determine optimum design of reinforced concrete cantilever retaining walls. Further explanations and applications can be found in recent books of Kaveh [19,20]. The objective function considered in this paper is the cost of the structure, and design is based on ACI 318-05 [21]. This function is minimized subjected to design constraints such as strength and stability constraints. A numerical example is presented in order to illustrate the performance of the utilized algorithms.

2. COLLIDING BODIES OPTIMIZATION ALGORITHM

The collision is a natural occurrence and the Colliding Bodies Optimization (CBO) algorithm was developed based on this phenomenon by Kaveh and Mahdavi [16]. In this technique, one object collides with other object and they move towards a minimum energy level. The CBO procedure can be briefly outlined as follows:

Step 1. The initial positions of the CBs are determined with random initialization of a population of individuals in the search space:

$$x_i^0 = x_{\min} + rand(x_{\max} - x_{\min}), \qquad i = 1,2,\ldots,2n \tag{1}$$

where x_i^0 determines the initial value vector of the ith CB. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable values vectors of variables; *rand* is a random number in the interval [0, 1]; and *n* is the number of CBs.

Step 2. The magnitude of the body mass for each CB is defined as:

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

where, $fit(i)$ represents the objective function value of the agent i; $2n$ is the population size. Obviously a CB with good value exerts a larger mass than the bad ones. Also, for maximizing the objective function, the term $\frac{1}{fit(i)}$ is replaced by $fit(i)$.

Step 3. The arrangement of the CBs objective function values is performed in ascending order (Fig. 1a). The sorted CBs are equally divided into two groups:

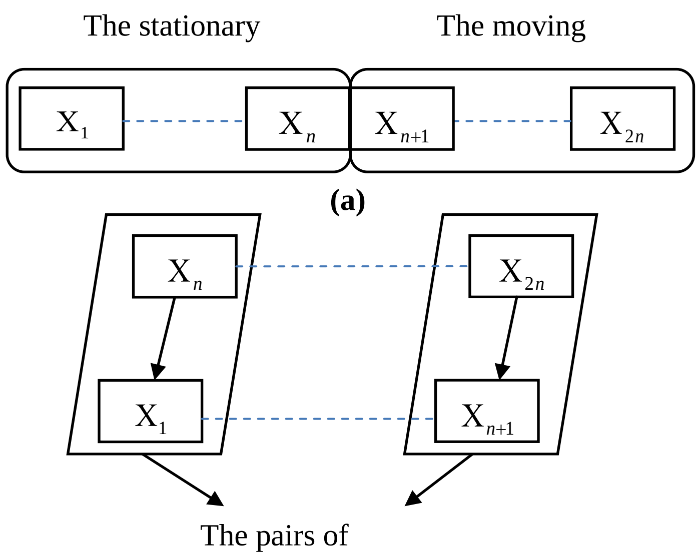

Figure 1. The sorted CBs in an increasing order. (b) The pairs of objects for the collision

The lower half of the CBs (stationary CBs) are good agents which are stationary and the velocity of these bodies before collision is zero. Thus:

$$v_i = 0, \qquad i = 1,2,\ldots,n \tag{3}$$

The upper half of CBs (moving CBs) move toward the lower half. Then, according to Fig. 1b, the better and worse CBs, i.e. agents with upper fitness value of each group will collide together. The change of the body position represents the velocity of these bodies before collision as:

$$v_i = x_{i-n} - x_i, \qquad i = n+1,\ldots,2n \tag{4}$$

where v_i and x_i are the velocity and position vectors of the ith CB in this group, respectively; x_i is the ith CB pair position of x_{i-n} in the previous group.

Step 4. After collision, the velocity of bodies in each group are evaluated using Eq. (5), Eq. (6) and the velocities before collision. The velocity of each moving CB after the collision is:

$$v_i' = \frac{(m_i - \varepsilon m_{i-n})v_i}{m_i + m_{i-n}}, \qquad i = n+1,\ldots,2n \tag{5}$$

where v_i and i v_i' are the velocity of the *i*th moving CB before and after the collision, respectively; *m* is the mass of the *i*th CB; m_{i-n} is mass of the *i*th CB pair. Also, the velocity of each stationary CB after the collision is:

$$v_i' = \frac{(m_{i+n} + \varepsilon m_{i+n})v_{i+n}}{m_i + m_{i+n}}, \qquad i = 1,2,\ldots,n \tag{6}$$

where v_{i+n} and v_i' are the velocity of the *i*th moving CB pair before and the *i*th stationary CB after the collision, respectively; m_i is mass of the *i*th CB; m_{i+n} is mass of the *i*th moving CB pair. As mentioned previously, ε is the coefficient of restitution (COR) and for most of the real objects, its value is between 0 and 1. It defined as the ratio of the separation velocity of two agents after collision to the approach velocity of two agents before collision. In the present algorithm, this index is used to control of the exploration and exploitation rate. For this goal, the COR is decreases linearly from unit to zero. Thus, ε is defined as:

$$\varepsilon = 1 - \frac{iter}{iter_{\max}} \tag{7}$$

where, $iter$ is the actual iteration number and $iter_{\max}$ is the maximum number of iterations, with COR being equal to unit and zero representing the global and local search, respectively [16].

Step 5. New positions of CBs are obtained using the generated velocities after the collision in position of stationary CBs.

The new positions of each moving CB is:

$$x_i^{new} = x_{i-n} + rand \circ v_i', \qquad i = n+1,\ldots,2n \tag{8}$$

where x_i^{new} and v_i' are the new position and the velocity after the collision of the *i*th moving CB, respectively; x_{i-n} is the old position of the *i*th stationary CB pair. Also, the new positions of stationary CBs are obtained by:

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

where x_i^{new} , x_i and v_i' are the new position, old position and the velocity after the collision of the *i*th stationary CB, respectively. $rand$ is a random vector uniformly distributed in the range $(-1,1)$ and the sign ‘‘$\circ$’’ denotes an element-by-element multiplication.

Step 6. The optimization process is repeated starting from Step 2 until the termination criterion, specified as the maximum number of iterations, is satisfied.

3. ENHANCED COLLIDING BODIES OPTIMIZATION ALGORITHM

In order to improve CBO to obtain faster and more reliable solutions, Enhanced Colliding Bodies Optimization (ECBO) was developed by Kaveh and Ilchi Ghazan [17] which uses memory to save a number of historically best CBs and also utilizes a mechanism to escape from local optima. The steps of this technique are given as follows:

Step 1. Initialization

The initial positions of all the CBs are determined randomly in an *m*-dimensional search space. These are evaluated according to Eq. (1).

Step 2. Defining mass

The value of mass for each CB is evaluated according to Eq. (2).

Step 3. Saving

Colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values. In this step, the solution vectors saved in CM are added to the population, and the same numbers of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order.

Step 4. Creating groups

CBs are divided into two equal groups: (i) stationary group, (ii) moving group (Fig. 1a).

Step 5. Criteria before the collision

The velocity of stationary bodies before collision is zero (Eq. (2)). Moving objects move toward stationary objects and their velocities before collision are calculated by Eq. (3).

Step 6. Criteria after the collision

The velocities of the stationary and moving bodies are evaluated using Eqs. (4) and (5), respectively.

Step 7. Updating CBs

The new position of each CB is calculated by Eqs. (7) and (8).

Step 8. Escape from local optima

In this step, a parameter like ***Pro*** within (0, 1) is introduced and it is 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,…,2n) which is a random number uniformly distributed within (0, 1). If rn_i < ***Pro***, one dimension of the *i*th CB is selected randomly and its value is regenerated as follows:

$$x_{ij} = x_{j,\min} + random.(x_{j,\max} - x_{j,\min}) \qquad (10)$$

where x_{ij} is the *j*th variable of the *i*th CB. $x_{j,\min}$ and $x_{j,\max}$ respectively, are the lower and upper bounds of the *j*th variable. In order to protect the structures of CBs, only one dimension is changed.

Step 9. Terminating condition check

The optimization process is terminated after a fixed number of iterations. If this criterion is not satisfied go to Step 2 for a new round of iteration process.

4. VIBRATING PARTICLES SYSTEM OPTIMIZATION ALGORITHM

The VPS is a population-based algorithm that simulates a free vibration of single degree of freedom systems with viscous damping (Kaveh and Ilchi Ghazaan [18]).The VPS has a number of particles consisting of the variables of the problem. The solution candidates gradually approach to their equilibrium positions which are achieved from current population and historically best position in order to have a proper balance between diversification and intensification. In VPS, the initial locations of particles are created randomly in an *n*-dimensional search space as

$$x_i^j = x_{\min} + rand.(x_{\max} - x_{\min}), \quad i = 1,2,...,n \tag{11}$$

where x_i^j is the *j* th variable of the particle *i*. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable variables vectors; *rand* is a random number uniformly distributed in the range of [0, 1].

For each particle, three equilibrium positions with different weights are defined, and during each generation, the particle position is updated by learning from them: (i) the historically best position of the entire population (HB), (ii) a good particle (GP), and (iii) a bad particle (BP). In order to select the GP and BP for each candidate solution, the current population is sorted according to their objective function values in an increasing order, and then GP and BP are chosen randomly from the first and second half, respectively.

A descending function based on the number of iterations is proposed in VPS to model the effect of the damping level in the vibration.

$$D = \left(\frac{iter}{iter_{max}}\right)^{-\alpha} \tag{12}$$

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

process. α is a constant.

According to the above concepts, the update rules in the VPS are given by

$$x_i^j = w_1.[D.A.rand1 + HB^j] + w_2.[D.A.rand2 + GP^j] + w_3.[D.A.rand3 + BP^j], \tag{13}$$

$$A = [w_1.(HB^j - x_i^j)] + [w_2.(GP^j - x_i^j)] + [w_3.(BP^j - x_i^j)], \tag{14}$$

$$w_1 + w_2 + w_3 = 1 \tag{15}$$

where x_i^j is the *j*th variable of the particle *i*. w_1, w_2, and w_3 are three parameters to measure the relative importance of HB, GP and BP, respectively. *rand1*, *rand2*, and *rand3* are random numbers uniformly distributed in the range of [0, 1].

In order to have a fast convergence in the VPS, the effect of BP is sometimes considered in updating the position formula. Therefore, for each particle, a parameter like p within (0,1) is defined, and it is compared with rand (a random number uniformly distributed in the range of [0,1]) and if p < rand, then $w_3 = 0$ and $w_2 = 1 - w_1$.

5. OPTIMAL DESIGN PROCESS

In this article, the design process starts by selecting upper and lower bound for design variables. Then the loads are applied on the retaining wall and both CBO and ECBO algorithms check the retaining wall for stability and if the dimensions satisfy stability criteria, the algorithms determine the moments and shear forces in critical sections and then calculate the required reinforcement and check the strength.

The design process of CBO algorithm consists of 6 steps while the ECBO algorithm consists of 9 steps. Both of these were explained in Sections 2 and 3, respectively.

5.1 Design variables of the problem

In this study, the model of the structure comprises of seven design variables that define the geometry of the cantilever retaining wall. These variables are illustrated in Fig. 2. The variables consist of the thickness of top stem (T1), the thickness of key and stem (T2), the toe width (T3), the heel width (T4), the height of top stem (T5), the footing thickness (T6), and the key depth (T7).

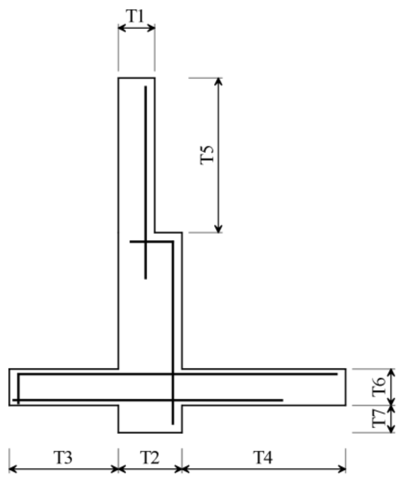

Figure 2. Schematic of the cantilever retaining wall and the considered design variables

4.2 Objective function

In this study, an objective function is defined as the total cost of the cantilever retaining wall. The cost minimization objective function includes the cost of concrete and reinforcing steel (both include the cost of the material per unit volume and costs associated with labor and installation). This objective function can be expressed as follows:

$$Q = V_{conc} \times (C_1 + C_2) + W_{steel} \times (C_3 + C_4) \tag{16}$$

By considering $\overline{Q} = Q/(C_1 + C_2)$, we have:

$$\overline{Q} = V_{conc} + W_{steel}\left(\frac{C_3 + C_4}{C_1 + C_2}\right) \tag{17}$$

where V_{conc} and W_{steel} are the volume of concrete and the weight of reinforcement steel in the unit of length (ft^3/ft or m^3/m, lb/ft or kg/m), C_1 and C_3 are the cost of the concrete and steel ($\$/lb$ or $\$/kg$), C2 and C4 are the cost of concreting and erecting reinforcement ($\$/lb$ or $\$/kg$).

Experience shows the value of $\frac{C_3 + C_4}{C_1 + C_2}$ can be in the range of 0.035–0.045 for different countries as indicated by Ref. [10], and in this paper 0.04 is selected.

The constraints of this problem are considered as follows:

$$g_1 = 1.5 - FS_{(overturning)} \leq 0 \tag{18}$$

$$g_2 = 1.5 - FS_{(sliding)} \leq 0 \tag{19}$$

$$g_3 = 2 - FS_{(bearing\,capacity)} \leq 0 \tag{20}$$

$$g_4 = M_u/(\phi_b M_n) - 1 \leq 0 \tag{21}$$

$$g_5 = V_u/(\phi_v V_n) - 1 \leq 0 \tag{22}$$

where, $FS_{(bearing\,capacity)}, FS_{(sliding)}$ and $FS_{(bearing\,capacity)}$ are the safety factors of overturning moment and sliding shear and bearing capacity of soil respectively; M_u and V_u are ultimate flexure and shear respectively; M_n and V_n are flexure and shear capacity respectively; ϕ_b and ϕ_v are strength reduction factor for flexure and shear, respectively.

Here, g_1 and g_2 refer to the constraints which are about stability of the cantilever retaining wall, and g_4 and g_5 refer to the constraints which are about shear and flexural strength. AASHTO [22] permits the factors of safety against sliding and overturning failure under dynamic loading condition reduced to 75% of the factors of safety used for the static loading designs.

A penalty function is used to enforce the constraints g_i on the objective function. The total objective function penalty $f_{prnalty}$ is a function of the summation of the stability, capacity, reinforcement configuration defined as:

$$f_{penalty}(x) = \left(1 + \sum_{i=1}^{n} \max(0, g_i(x))\right)^k \tag{23}$$

In which *n* is the number of constraints and *k* is selected considering the exploitation rate of the search space (typically > 1). The penalized objective function $Merit(x)$ is a product of either the cost objective function of candidate design *i* and its total penalty as

$$Merit(x) = \overline{Q} \times f_{penalty}(x) \tag{24}$$

The penalty function imposes a numerical penalty on the value of the objective function that tends to reflect the degree at which the constraints are violated by a candidate set of design variables [9].

6. NUMERICAL EXAMPLE

The process of optimization was described in section 4. For this purpose a computer program is written in Matlab for analysis, design and optimization. The analysis and design are in the form of a function which is called by the optimization program.

In this article, two backfills are considered and properties of these backfills are presented in Table 1. These properties are includ soil specific weight (γ_b) and internal friction angle of soil (φ). Design is based on 1.0m wide strip of the retaining wall. The data for the considered retaining wall are designed with two types of soil as follow:

- The total height of stem is constant and equal to 6.1*m*
- Ground water level is assumed to be below the foundation level of the wall and therefore not affecting the soil properties.
- Surcharge load is 10 kN/m^2
- The 28 days concrete cylinder strength is 25*MPa* and concrete specific weight is 24 kN/m^3
- Rebar yield stress is 300*MPa* and specific weight of ribbed bar is 78 kN/m^3
- The allowable soil pressure is taken as q_a =300 kN/m^2 (3 kg/cm^2).
- The h_p is equal to zero
- The clear concrete cover is 50 mm.
- Wall friction angle (δ) according to ACI 318-05[20] is δ=tan(φ)

- The number of agents and number of iterations in both of algorithms is 20 and 300, respectively.

The static design is performed for both F1 and F2 backfills, utilizing Coulomb theories. Upper and lower bounds for the design variable are shown in Table 2. Critical sections are illustrated in Fig. 3. A schematic view of a concrete retaining wall is illustrated in Fig. 4**.**

Table 1: Types of the backfills considered in the present work

	Description	**Density (kN/m^3)**	**Internal friction angle (°)**
F1	Coarse granular fills (GW,GP)	22	35
F2	Granular soils with more than 12% of fines (GW, GS, SM, SL) and fine soils with more than 25% of coarse grains (CL–ML)	20	30

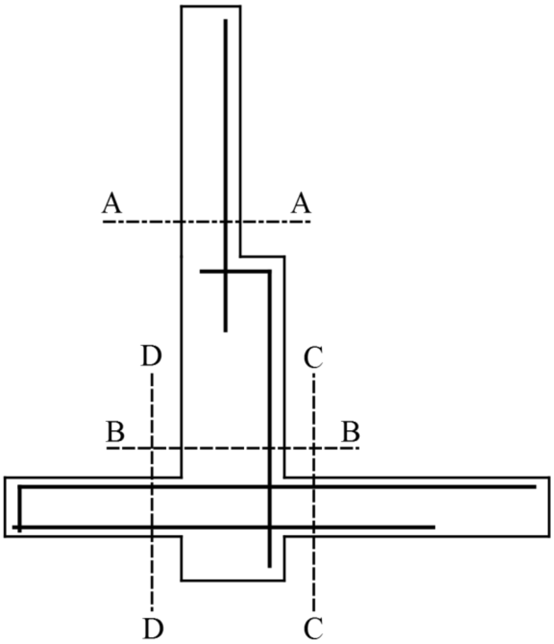

Figure 3. Critical sections of the wall

Table 2: Upper and lower bounds for design variables

Design variables	**Thickness of the top stem (cm)**	**Thickness of the key (cm)**	**Toe width (cm)**	**Heel width (cm)**	**Height of the top stem (cm)**	**Footing thickness (cm)**	**Key depth (cm)**
Lower bound	30	30	45	180	150	30	20
Upper bound	60	60	120	300	610	90	90

Results of optimization of each soil type are presented for CBO and ECBO in this study and other researchers in Tables 3 and 4. Histories of the optimization for soil type F1 and F2 are also depicted in Fig. 5 and Fig. 6, respectively. These figures show that the ECBO finds better fitness in comparison to CBO for design of retaining wall and the rapid convergence of the ECBO is considerable. Also results of 20 independent run for F1 and F2 soil types are illustrated in Fig. 7 and Fig. 8, respectively. These figures show that optimal solutions in the ECBO algorithm has lower standard deviation than CBO algorithm in 20 independent run. Standard deviation of CBO (std≈0.03) is 100 time bigger than ECBO (std≈0.003). According to these figures, 40% of the solutions of the two algorithms are close to each other.

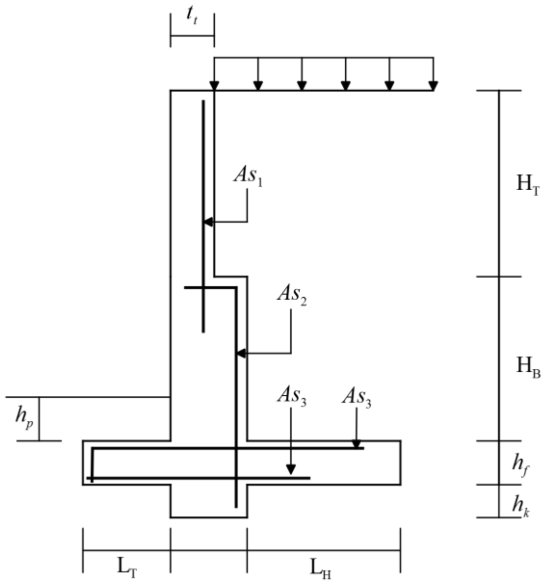

Figure 4. Schematic view of a RC cantilever retaining wall

Table 3: Optimum results for F1 back fill type

Design variables	**Other algorithms**				**Present work**		
	HIS[10]	**CSS[11]**	**DPSO[14]**	**DEO[15]**	**CBO**	**ECBO**	**VPS**
T_1(m)	0.33	0.31	0.31	0.30	0.32	0.30	0.30
T_2(m)	0.60	0.36	0.60	0.58	0.56	0.56	0.56
T_3(m)	1.20	1.01	1.18	1.19	0.98	0.98	0.98
T_4(m)	2.56	2.40	1.85	1.83	1.80	1.80	1.80
T_5(m)	3.25	4.13	1.50	3.38	3.80	3.61	3.61
T_6(m)	0.57	0.34	0.33	0.40	0.35	0.35	0.35
T_7(m)	0.67	0.39	0.23	0.21	0.20	0.2	0.2
As_1(mm^2/m)	1033.0	1093.5	1218.0	1167.5	1203.2	1120.1	1120.0
As_2(mm^2/m)	3000.0	2083.3	2566.0	2818.1	2318.5	2319.2	2318.5
As_3(mm^2/m)	2653.0	2574.8	1468.0	2599.7	2383.1	2447.6	2443.1
As_4(mm^2/m)	1054.0	1200.6	1344.0	1537.3	1365.8	1336.5	1338.9

Table 4: Optimum results for F2 back fill type

Design variables	Other algorithms				Present work		
	HIS [10]	CSS [11]	DPSO [14]	DEO [15]	CBO	ECBO	VPS
T_1(m)	0.34	0.31	0.34	0.31	0.30	0.30	0.30
T_2(m)	0.60	0.43	0.59	0.59	0.59	0.59	0.59
T_3(m)	1.17	1.06	1.18	1.16	1.20	1.12	1.13
T_4(m)	2.13	2.71	2.15	2.62	1.80	1.80	1.80
T_5(m)	3.33	4.49	1.50	3.32	3.43	3.43	3.44
T_6(m)	0.56	0.30	0.30	0.51	0.34	0.34	0.34
T_7(m)	0.35	0.61	0.22	0.43	0.20	0.20	0.20
As_1(mm^2/m)	926.0	1688.2	1338.0	1216.4	1120.0	1120.0	1120.1
As_2(mm^2/m)	2634.0	2218.0	3216.0	3158.3	2468.6	2468.8	2468.6
As_3(mm^2/m)	2148.0	2941.5	4210.0	2868.8	2491.4	2513.1	2537.6
As_4(mm^2/m)	1034.0	1312.6	1166.0	1044.3	1298.4	1300.7	1289.3

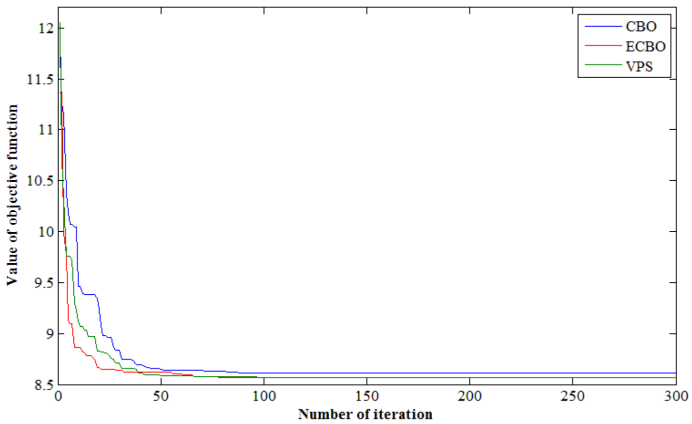

Figure 5. Optimization histories for F1 backfill type

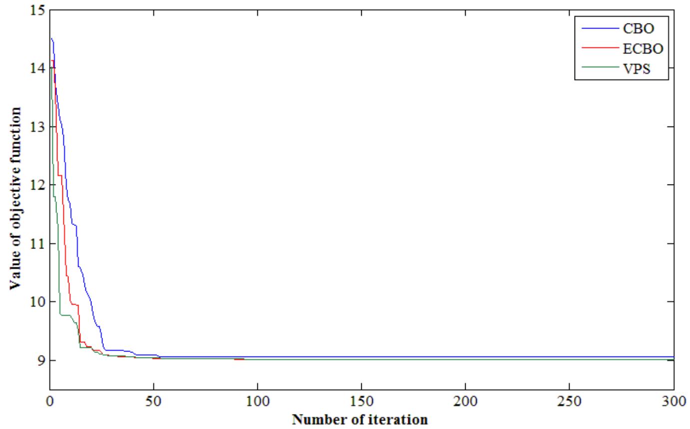

Figure 6. Optimization histories for F2 backfill type

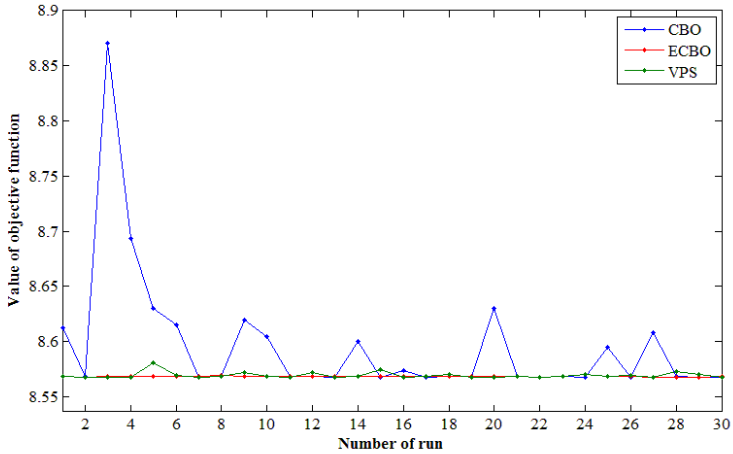

Figure 7. Results of 20 independent run for F1 back fill type

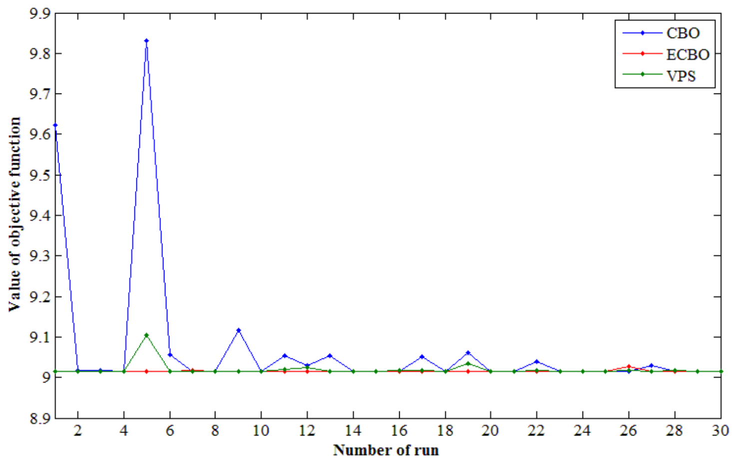

Figure 8. Results of 20 independent run for F2 back fill type

7. CONCLUDING REMARKS

In this study, CBO, ECBO and VPS algorithms are utilized to optimize RC retaining walls in continuous search space. The CBO has simple structure and requires no internal parameter tuning and does not use memory for saving the best so far solutions. The ECBO uses memory to save a number of historically best CBs and also utilizes the random perturbation mechanism to update the positions. VPS has a number of individuals consisting of the variables of the problem. The solution candidates gradually approach to their equilibrium positions that are achieved from current population and historically best position in order to have a proper balance between diversification and intensification.

Design of the retaining wall is based on ACI code. The algorithms are applied to an objective function that is the total cost of retaining wall. Optimal solutions of CBO, ECBO and VPS algorithms and results of other algorithms such as DPSO and DEO are presented in Tables 3 and 4. Optimization histories of the CBO, ECBO and VPS algorithms are compared. Generally, the VPS has better performance than ECBO and CBO and other methods in terms of accuracy and speed of convergence.

REFERENCES

1. Das BM. *Principles of Foundation Engineering*, Cengage Learning, 2015.
2. Rhomberg EJ, Street WM. Optimal design of retaining walls, *Journal of the Structural Division*, *ASCE* **107**(1981) 992-1002.
3. Saribas A, Erbatur F. Optimization and sensitivity of retaining structures, *Journal of Geotechnical Engineering*, **122**(1996) 649-56.
4. Ceranic B, Fryer C, Baines R. An application of simulated annealing to the optimum design of reinforced concrete retaining structures, *Computers and Structures*, **79**(2001) 1569-81.
5. Castillo E, Mínguez R, Ruiz Terán A, Fernández-Canteli A. Design and sensitivity analysis using the probability-safety-factor method, An application to retaining walls, *Structural Safety*, **26**(2004) 159-79.
6. Basudhar P, Lakshman B, Dey A. Optimal cost design of cantilever retaining walls, *India: IGC, Chennai*, (2004) 14-6.
7. Sivakumar Babu G, Basha BM. Optimum design of cantilever retaining walls using target reliability approach, *International Journal of Geomechanics*, **8**(2008) 240-52.
8. Yepes V, Alcala J, Perea C, González-Vidosa F. A parametric study of optimum earth-retaining walls by simulated annealing, *Engineering Structures*, **30**(2008) 821-30.
9. Camp CV, Akin A. Design of retaining walls using Big Bang - Big crunch optimization, *Journal of Structural Engineering*, **138**(2011) 438-48.
10. Kaveh A, Shakouri Mahmud Abadi A. Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls, *International Journal of Civil Engineering*, **9**(2011) 1-8.
11. Kaveh A, Behnam A. Charged system search algorithm for the optimum cost design of reinforced concrete cantilever retaining walls, *Arabian Journal for Science and Engineering*, **38**(2013) 563-70.
12. Kaveh A, Kalateh-Ahani M, Fahimi-Farzam M. Constructability optimal design of reinforced concrete retaining walls using a multi-objective genetic algorithm, *Structural Engineering and Mechanics*, **47**(2013) 227-45.
13. Kaveh A, Khayatazad M. Optimal design of cantilever retaining walls using ray optimization method, *Transactions of Civil Engineering*, **38**(2014) 261-74.
14. Kaveh A, Soleimani N. CBO and DPSO for optimum design of reinforced concrete cantilever retaining walls, *Asian Journal of Civil Engineering (BHRC)*, **16**(2015) 751-74.
15. Kaveh A, Farhoudi N. Dolphin echolocation optimization for design of cantilever retaining walls, *Asian Journal of Civil Engineering (BHRC)*, **17**(2016) 193-211.
16. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method,

Computers and Structures, **139**(2014) 18-27.
17. Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies optimization for design problems with continuous and discrete variables, *Advances in Engineering Software*, **77**(2014) 66-75.
18. Kaveh A, Ilchi Ghazaan M. A new meta-heuristic algorithm: vibrating particles system, *Acta Mechanica*, Doi: 10.1007/s00707-016-1725-z.
19. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer, Switzerland, 2nd Edition, 2017.
20. Kaveh A. *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*, Springer, Switzerland, 2017.
21. A.C. Institute, *Building code requirements for structural concrete (ACI 318-05) and commentary (ACI 318R-05)*: American Concrete Inst, 2004.
22. A.A.O.S. Highway, T. Officials, *Standard specifications for highway bridges*: AASHTO, 2002.