ASIAN JOURNAL OF CIVIL ENGINEERING (BHRC) VOL. 15, NO. 6 (2014)
PAGES 788-802

OPTIMUM COST DESIGN OF REINFORCED CONCRETE ONE-WAY RIBBED SLABS USING CBO, PSO AND DEMOCRATIC PSO ALGORITHMS

A. Kaveh* and Sh. Bijari
Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering, Iran University of Science and Technology, Tehan-16, Iran

Received: 10 February 2014; **Accepted:** 18 June 2014

ABSTRACT

The main objective of the current study is to utilize the capabilities of recently developed meta-heuristic algorithms for structural cost optimization of a one-way reinforced concrete ribbed slab simply supported at both ends. Two of these new and simple optimization algorithms, known as colliding bodies optimization (CBO) and democratic particle swarm optimization (DPSO), and a renowned optimization algorithm, PSO, are presented to solve cost optimization of a concrete ribbed slab. Although PSO is a very well-known and commonly used optimization algorithm, democratic PSO is an improved version of particle swarm optimization method. In DPSO the emphasis is placed upon improving the premature convergence phenomenon which is believed to be one of defects of the original PSO. CBO utilizes simple formulation to find optimum values and does not need any internal parameter. Performance of these algorithms is compared with harmony search. The results illustrate the power of the CBO and effectiveness of improvements of DPSO method in the present optimization problem.

Keywords: Reinforced concrete slab; one-way joist floor; particle swarm optimization; democratic particle swarm optimization; colliding bodies optimization; cost optimization.

1. INTRODUCTION

Meta-heuristics algorithms are recent generation of the optimization approaches to solve complex problems. These methods explore the feasible region based on both randomization and some specified rules through a group of search agents [1]. Laws of natural phenomena are usually source of the rules. Genetic Algorithm (GA) is introduced by Holland [2] and Goldberg [3]. It is inspired by biological evolutions theory. Particle swarm optimization (PSO) is introduced by Eberhart and Kennedy [4]. It simulates social behavior, and it is inspired by

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

the migration of animals in a bird flock or fish school. The particle swarm algorithm is applied to truss optimization with dynamic constraints [5]. Ant Colony Optimization (ACO) is presented by Dorigo et al. [6]. It imitates foraging behavior of ant colonies. There are several other natural-inspired algorithms such as Simulated Annealing (SA) presented by Kirkpatrick et al. [7], Harmony Search (SA) introduced by Geem et al. [8], Big Bang–Big Crunch algorithm (BB–BC) presented by Erol and Eksin [9], and improved by Kaveh and Talatahari [10]. Due to good performance of these algorithms and their simple implementation, they have been widely used for solving various problems in different fields of science and engineering. One of the most recent meta-heuristic algorithms is the Charged System Search (CSS) proposed by Kaveh and Talatahari [11]. Electric laws of physics and the Newtonian laws of mechanics are used for guiding the Charged Particles (CPs) to explore the locations of the optimum. The scenario in CSS is completed by the addition of magnetic forces in Magnetic Charged System Search (MCSS) method, Kaveh et al. [12].

Particle Swarm Optimization (PSO) initially proposed by Kennedy and Eberhart [4] is one of the most widely used population-based meta-heuristic algorithms. This method performs easily in engineering problems, needs little number of parameters and has high power in finding suboptimal solutions in a reasonable amount of time. So many researchers are continually encouraged in using PSO for a varied range of optimization problems in different disciplines. In structural engineering, PSO has been successfully applied to diverse types of optimization problems ([18–24] among others). However, in spite of having the above-referred advantages, the standard PSO is infamous of premature convergence [25,26]. One of active research topics in recent years is improving the exploration ability of the PSO [27].

Democratic Particle Swarm Optimization (DPSO) proposed by Kaveh and Zolghadr [28] improves the exploration capabilities of the PSO and thus addresses the problem of premature convergence. In accordance to the algorithm name, Democratic PSO, all eligible particles have the right to be involved in decision making in this algorithm. The details of the method will be represented in the following sections.

The Colliding Bodies Optimization (CBO) developed by Kaveh and Mahdavi [29] is an efficient and simple algorithm based on one-dimensional collisions between two bodies, where each agent solution is modeled as the body. This algorithm utilizes simple formulation, and it requires no parameter tuning [30]. The details of the method will be represented in the following sections.

A one-way joist floor system comprises of hollow slabs which depth is more than solid slabs. For buildings with the small superimposed loads and the relatively large spans this system is the most economical such as in schools, hospitals, and hotels. Since the concrete in the tension zone is ineffective; this region is kept open between the ribs or filled with lightweight material to reduce the slab weight.

In this paper Colliding Bodies Optimization, standard Particle Swarm Optimization and democratic Particle Swarm Optimization are utilized for optimal design of a concrete ribbed slab. Comparison of the convergence curves of these methods with that of the HS algorithms demonstrate that the CBO and DPSO methods are powerful and efficient approaches for finding the optimum solution to structural optimization problems. In this example, the CBO and DPSO performed meaningfully better than the HS and PSO by attaining the best solutions so far.

The remainder of this paper is organized as follows: In Section 2, problem statement, objective function and design constraints is presented. In Section 3, three optimization algorithms, particle swarm optimization, democratic particle swarm optimization and colliding bodies optimization algorithms are briefly presented. One numerical example of a one-way reinforced concrete ribbed slab is studied in Section 4. Conclusion is provided in Section 5.

2. PROBLEM STATEMENT

In a reinforced concrete one-way ribbed slab optimization problem the aim is to minimize the cost of the structure while satisfying some constraints. To model the ribbed slab, six discrete design variables are considered as shown in Fig. 1. These contain the thickness of the top slab (D_1), the rib spacing (D_2), the rib width at the lower end (D_3), the rib width at the top end (D_4), the bar diameter (D_5), and the rib depth (D_6).

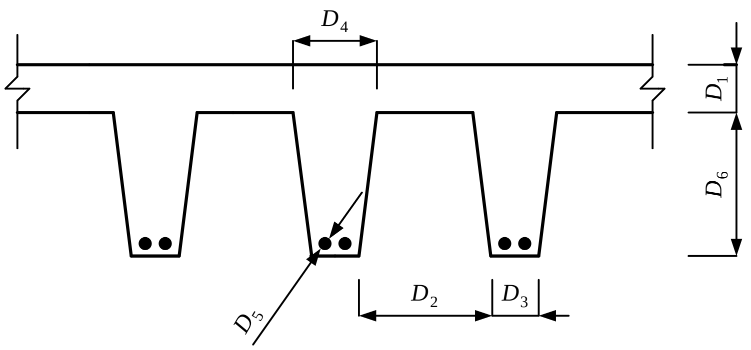

Figure 1. A ribbed slab and the design variables selected

2.1 Optimum Design Process

Typical design of the ribbed slabs consists of two phases:

(1) Selecting random values for the variables and checking the dimensions according to the ACI 318.08 standard [31].
(2) Calculating the required reinforcement and checking the strength.

2.2 Objective function

The objective function of concrete ribbed slab optimization includes the costs associated with concrete and steel material as well as concreting and erecting the reinforcement which must be minimized. This can be achieved by determining the optimal values for decision variables D_1 to D_6. The objective function can be expressed as follows:

$$Cost = V_{conc} \times (C_{cm} + C_c) + W_{steel} \times (C_{sm} + C_e) \tag{1}$$

Considering $\overline{Cost} = Cost / (C_{cm} + C_c)$, we have:

$$\text{Minimize } \overline{Cost} = (V_{conc} + W_{steel}(\frac{C_{sm} + C_e}{C_{cm} + C_c}))/l \tag{2}$$

where V_{conc} and W_{steel} are the volumes of the concrete and the weight of the reinforcement steel in the unit length (m^3/m, kg/m), respectively; C_{cm} and C_{sm} are the costs of concrete and steel ($/kg for steel and $/$m^3$ for concrete), respectively; C_c and C_e are the costs of concreting and erecting the reinforcement, respectively. l is the center-to-center distance of the ribs. Based on reviews and the cost estimation performed, a value of 0.04 for the coefficient C $(C = (\frac{C_{sm} + C_e}{C_{cm} + C_c}))$ is obtained.

2.3 Design Constraints

For designing this problem according to the ACI 318-08 [31] the following constraints must be considered.

2.3.1 Flexural Constraint

The flexural constraint can be described in the following form:

$$M_u / (\varphi_b M_n) \leq 1 \tag{3}$$

where M_u and M_n are the ultimate design moment and the nominal bending moment, respectively.

2.3.2 Shear Constraint

The shear constraint can be described as:

$$V_u / (\varphi_v V_n) \leq 1 \tag{4}$$

where V_u and V_c are the ultimate factored shear force and the nominal shear strength of the concrete, respectively. The concrete should carry the total shear because no stirrup is used in the slab. The shear strength V_c provided by the concrete for the ribs may be taken to be 10% greater than that of the beams. This is mainly due to the interaction between the slab and the closely spaced ribs.

2.3.3 Serviceability Constraints

The serviceability constraints are expressed in terms of the limits on the steel reinforcement ratio and the bar spacing. The steel reinforcement ratio should satisfy the following constraint:

$$\rho \leq \rho_{\max} = 0.75\rho_b \tag{5}$$

The minimum shrinkage steel ratio, $\rho_{\min}$, in the slab is 0.002 for slabs in which bars of grade 40 or 50 are utilized and 0.0018 for slabs in which deformed bars of grade 60 are used. The bar spacing should satisfy the following constraints:

• The minimum clear spacing between parallel bars in a layer, db, should not be less than 25 mm.

• The maximum spacing between the bars ≤ 5 times the rib thickness ≤ 450 mm (18 in.).

2.3.4 Deflection Constraints

The thickness of the top slab should not be less than 1/12 of the clear span between the ribs or 50 mm (2 in.). Based on the ACI code [31] a minimum slab thickness $h_{\min}$ of L/16, L/18.5, L/21, or L/8 is required, depending on the support conditions. Here, L is the effective span length of the slab.

2.3.5 Other Constraints

The ribs should not be less than 100 mm in width, and should have a depth of no more than 3.5 times the minimum width of the rib. Clear spacing between the ribs should not exceed 750 mm. A limit on the maximum spacing of the ribs is required because of the special provisions permitting higher shear strengths and lower concrete protection for the reinforcement of these relatively small repetitive members.

3. OPTIMIZATION ALGORITHMS

For making the improvements visible a basic form of the algorithm which is mentioned here as the standard PSO will be briefly summarized first. Since PSO has been gradually improved by different researchers and for better comparison the description of the standard PSO is reproduced from Ref. [32].

3.1 Particle Swarm Optimization Algorithm

Particle Swarm Optimization, first developed by Kennedy and Eberhart [4], is a population-based meta-heuristic algorithm inspired by the social behavior of animals such as fishes schooling, insects swarming, and birds flocking. Like many other population-based meta-heuristic algorithm, PSO begins with a set of particles which are randomly spread in the multi-dimensional search space of problem. These particles are supposed as potential solutions of the optimization problem at hand. By an objective function the fitness of each candidate solution is measured. As the optimization process develops these particles move around in the search space searching for better positions. By gradual improvement of the positions of the particles in a population the algorithm finally converges to a sub-optimal solution.

In PSO to find better positions, particles utilize two different resources of information: their own best experience which is called a local best position and the swarm's best position so far which is called the global best position. Based on these two types of information, a particle makes a decision about the next position it is going to experience in iteration (k + 1) by forming a velocity vector as follows:

$$v_{i,j}^{k+1} = \chi\left[\omega v_{i,j}^{k} + c_1 r_1\left(xlbest_{i,j}^{k} - x_{i,j}^{k}\right) + c_2 r_2\left(xgbest_{j}^{k} - x_{i,j}^{k}\right)\right] \quad (6)$$

where, $v_{i,j}^{k}$ is the velocity or the amount of change of the design variable *j* of particle *i*, $x_{i,j}^{k}$ is the current value of the *j*th design variable of the *i*th particle, $xlbest_{i,j}^{k}$ is the best value of the design variable *j* ever found by *i*th particle, $xgbest_{j}^{k}$ the best value of the design variable *j* experienced by the entire swarm so far, r_1 and r_2 are two random numbers uniformly distributed in the range (1,0), c_1 and c_2 are two parameters representing the particle's confidence in itself and in the swarm, respectively. In this paper, these parameters which determine the particle's inclination to move toward local and global best experiences are taken as 2 as reported to be suitable in Ref. [33], however these had been taken as 1.5 in Ref. [32]. Here, w is the inertia weight for the previous iteration's velocity and it can be set in order to control the exploration of the algorithm. In Ref. [32] this parameter is defined as:

$$w = 0.4[1 + min(cov, 0.6)] \quad (7)$$

where *cov* is the coefficient of variation of the swarm's objective function. The parameter χ is used to avoid divergence behavior and can be obtained from the following expression as indicated by [34]:

$$\chi = \frac{1.6}{\left|2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)}\right|} \quad (8)$$

Once the velocity vector is defined, the new positions of the particles are determined as:

$$x_{i,j}^{k+1} = x_{i,j}^{k} + v_{i,j}^{k+1} \quad (9)$$

where the time interval is considered as unity, so that the addition of the velocity vector to the position vector becomes permissible.

3.2 Democratic Particle Swarm Optimization Algorithm

Democratic PSO algorithm introduced by Kaveh and Zolghadr [28] is an improved version of Particle Swarm Optimization algorithm and improves the exploration capabilities of the PSO. In this algorithm the main objective is to decrease the premature convergence which is believed to be one of the main weaknesses of PSO.

Democratic PSO like PSO is a population-based meta-heuristic algorithm. In the democratic PSO for improving PSO the next position of a particle is determined based on the attitude of a bigger set of members called eligible members. In DPSO the particles can get their information from a more diverce set of resources. Moreover, letting particles with seeming lower quality to take part in decision provides the algorithm a better exploration characteristics. Based on this information, a particle makes a decision about the next position it is going to experience in the (*k*+1)th iteration by forming a velocity vector as follows:

$$v_{i,j}^{k+1} = \chi\left[\omega v_{i,j}^{k} + c_1 r_1\left(xlbest_{i,j}^{k} - x_{i,j}^{k}\right) + c_2 r_2\left(xgbest_{j}^{k} - x_{i,j}^{k}\right) + c_3 r_3 d_{i,j}^{k}\right] \quad (10)$$

in which $d_{i,j}^{k}$ is the *j*th variable of the vector *D* for the *i*th particle. The vector *D* represents the democratic effect of the other particles of the swarm on the movement of the *i*th particle. r_3 is a random number uniformly distributed in the range (1,0). Parameter c_3 is introduced to control the weight of the democratic vector. Here, the vector *D* is taken as:

$$D_i = \sum_{k=1}^{n} Q_{ik}\left(X_k - X_i\right) \quad (11)$$

where Q_{ik} is the weight of the *k*th particle in the democratic movement vector of the *i*th particle and can be defined as:

$$Q_{ik} = \frac{E_{ik}\dfrac{obj_{best}}{obj(k)}}{\sum_{j=1}^{n} E_{ij}\dfrac{obj_{best}}{obj(j)}} \quad (12)$$

in which *obj* is the objective function value; obj_{best} is the value of the objective function for the best particle in the current iteration; *X* is the position vector of the particle; *E* is the eligibility parameter and is analogous to parameter *P* in CSS [35]. In a minimization problem *E* can be defined as:

$$E_{ik} = \begin{cases} 1 & \dfrac{obj(k) - obj(i)}{obj_{worst} - obj_{best}} > rand \vee obj(k) < obj(i) \\ 0 & else \end{cases} \quad (13)$$

where obj_{worst} and obj_{best} are the values of the objective function for the worst and the best particles in the current iteration, respectively. The symbol $\vee$ stands for the union.

Since a term is added to the velocity vector of the PSO, the parameter χ should be decreased in order to avoid divergence. Here, this parameter is determined using a trial and error process.

According to Eq. (12), all of the better particles and some of the particles with lower fitness values affect the new position of the particle under consideration. This modification increases the performance of the algorithm in two ways: (1) helps the particles to get information about good zones of the search space other than those experienced by themselves and the best particle of the population and (2) causes some bad particles to take part in the movement of the swarm and thus improving the exploration capabilities of the algorithm. Both of the above effects help to decrease the premature convergence of the algorithm.

3.3 Colliding Bodies Optimization Algorithm

Engineers and natural philosophers always inspire from nature, and many meta-heuristic approaches are inspired by solutions that nature herself appears to have selected for hard problems [35]. The collision is also a natural event, which happens between objects, bodies, cars, etc. The idea of the CBO algorithm is based on the study of a collision between two bodies in one-dimension; in which one body collide with other body and they move toward minimum energy level, [18].

3.3.1 The CBO algorithm

In CBO algorithm, the solution candidates, X_i, including a number of variables (i.e. $X_i = \{X_{i,j}\}$) are considered as colliding bodies (CBs). The massed objects are consisted of two main identical groups which are known as stationary and moving objects, where the moving objects move to pursue stationary objects and a collision happens between pairs of objects. This is done for two purposes: (i) to get better the positions of moving objects; (ii) to push stationary objects towards better positions. After the collision, the new positions of the colliding bodies are updated and the new velocity is obtained by two laws that govern collisions between bodies (i) laws of momentum and (ii) laws of energy [29].

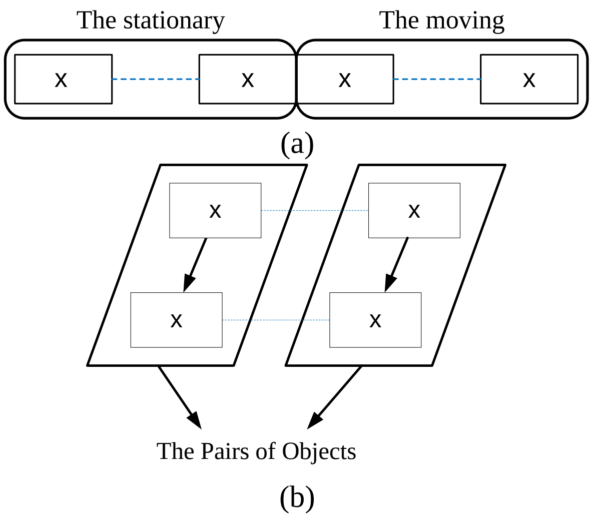

Figure 2. (a) The sorted CBs in an ascending order. (b) The pairs of objects for the collision

The CBO procedure can briefly be stated as follows:

(1) The initial positions of CBs are obtained with random initialization of a population of individuals in the search space and their associated values of the objective

function:

$$x_i^0 = x_{\min} + rand\,(x_{\max} - x_{\min}), \qquad i = 1, 2, ..., n \tag{14}$$

where x_i^0 determines the initial value vector of the *i*th 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.

(2) The value of the body mass for each CB is defined as:

$$m_k = \frac{\frac{1}{fit(k)}}{\sum_{i=1}^{n} \frac{1}{fit(i)}} \tag{15}$$

where *fit(i)* represents the objective function value of the agent *i*; *n* is the population size. Obviously a CB with good values 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)*.

(3) The ranking of the CBs objective function values is performed in an increasing order Fig. 2a. The sorted CBs are equally divided into two groups. Then, the pairs of CB are defined for collision:

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

$$v_i = 0, \qquad i = 1, 2, ..., \frac{n}{2} \tag{16}$$

- The upper half of CBs (moving CBs): These CBs move toward the lower half. Then, according to Fig. 2b, 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:

$$v_i = x_i - x_{i-\frac{n}{2}}, \qquad i = \frac{n}{2} + 1, ..., n \tag{17}$$

where v_i and x_i are the velocity and position vectors of the *i*th CB in this group, respectively; $x_{i-\frac{n}{2}}$ is the *i*th CB pair position of x_i in the previous group.

(4) After the collision, the velocity of bodies in each group is evaluated using the collision laws and the velocities before collision. The velocity of each moving CB after the collision is evaluated as following:

$$v_i' = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}}, \qquad i = \frac{n}{2}+1,\ldots,n \tag{18}$$

where v_i and v_i' are the velocity of the *i*th moving CB before and after the collision, respectively; m_i is the mass of the *i*th CB; $m_{i-\frac{n}{2}}$ is mass of the *i*th CB pair. Also, the velocity of each stationary CB after the collision is:

$$v_i' = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}}, \qquad i = 1,\ldots,\frac{n}{2} \tag{19}$$

$v_{i+\frac{n}{2}}$ 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; $m_{i+\frac{n}{2}}$ is mass of the *i*th moving CB pair. ε is the coefficient of restitution (COR) and for most of the real objects, its value is between 0 and 1. This coefficient is defined as the ratio of the separation velocity of two agents after collision to the approach velocity of two agents before collision. This index is used to control of the exploration and exploitation rate. For this goal, the COR is decreased linearly from unity to zero. Thus, ε is defined as:

$$\varepsilon = 1 - \frac{iter}{iter_{\max}} \tag{20}$$

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 [13].

(5) New positions of CBs are determined using the generated velocities after the collision in position of stationary CBs.

The new positions of each moving CB is:

$$x_i^{new} = x_{i-\frac{n}{2}} + rand \circ v_i', \qquad i = \frac{n}{2}+1,\ldots,n \tag{21}$$

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

$$x_i^{new} = x_i + rand \circ v_i', \qquad i = 1,...,\frac{n}{2} \tag{22}$$

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.

(6) The optimization is repeated starting from Step 2 until a termination criterion, specified as the maximum number of iteration, is fulfilled. It should be noted that, a body's status (stationary or moving body) and its numbering are changed in two subsequent iterations.

The CBO algorithm does not contain internal parameters except the coefficient of restitution (COR). By considering the linear variation law for COR, this algorithm becomes a parameter independent optimization approach. This is a definite power of the CBO.

4. NUMERICAL EXAMPLE

In this section to verify the efficiency of the algorithms, CBO, DPSO and PSO, and compare them with HS algorithm [36] an example of a one-way reinforced concrete ribbed slab simply supported at both ends is presented. The general data for the example is provided in Table 1. The design variables are presented in Table 2. A general plan of the concrete ribbed slab is illustrated in Fig. 3. The results of the optimum design are presented in Table 3, and the convergence curves are shown in Fig. 4.

Fig. 4 compares the convergence curves for the one-way reinforced concrete ribbed slab attained by the Harmony Search, Colliding Bodies Optimization, Democratic Particle Swarm Optimization and Standard Particle Swarm Optimization algorithms. Investigation of the convergence curves in Fig. 4 provides some useful points about the differences of the four algorithms. For this problem by HS algorithm in Ref. [36], the number of iteration 10000 and harmony memory size 30 are considered. In convergence curve of HS y-axis starts from 2.5 and x-axis end 10000, however, for a better observation y-axis and x-axis are bounded to 1.83 and 6000, respectively. Number of particle and iteration in this example for each three methods are 30 and 200, respectively. Convergence curve of Harmony Search in 3000th analysis became straight and the exploration is terminated. But as it can be seen from Fig. 4 the CBO reaches the final result in 1440th analyses. PSO's convergence curve shows that the convergence is obtained in the 1260 analyses and after that it became straight. On the other hand democratic PSO reached an initial convergence after 360th analyses and it still continued exploring the search space until it reached the final answer at 3480th analysis. This can be interpreted as the modifications being effective on the improvement of the premature convergence problem. It should be noted that the weight obtained by CBO, DPSO and PSO is much less than the weight obtained by HS, and the answers of CBO and DPSO are less than that of PSO.

Table 1: Common data for the considered example

f_y	420 MPa
$f_c^{'}$	28 MPa
DL	0.78 kN/m^2
LL	4 kN/m^2
L	6 m
Cover	20 mm
w_s	78.5 kN/m^3
w_c	24 kN/m^3

Table 2: Design variables

	Value (cm)
Slab thickness	2.5, 5, 7.5, 10
Rib spacing	40, 42.5, 45,…, 72.5, 75
Rib width at lower end	10, 12.5,…, 22.5, 25
Rib width at taper end	10, 12.5,…, 27.5, 30
Bar diameter	1, 1.2, 1.4, 1.6, 1.8, 2
Rib depth	15, 17.5,…, 72.5, 75

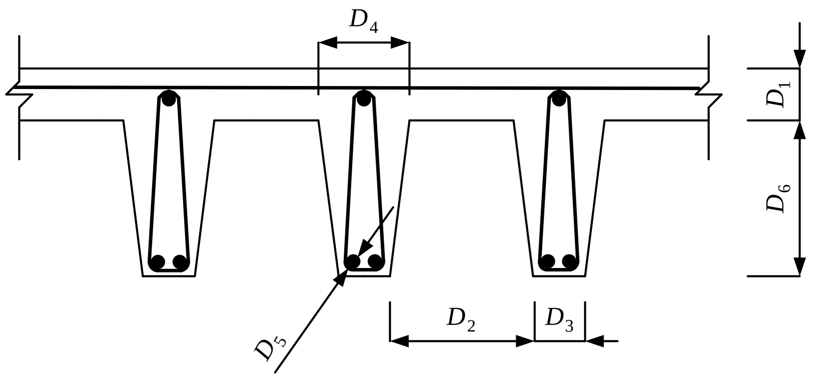

Figure 3. General plan of a concrete ribbed slab

Table 3: Results of the optimization

Algorithm	Slab thickness (cm)	Rib spacing (cm)	Rib width at lower end (cm)	Rib Width at taper end (cm)	Bar diameter (cm)	Rib depth (cm)	Weight ($/m^2)	Number of analyses
HS	5	60	10	10	1.4	35	1.3626	6000
PSO	5	60	17.5	17.5	1.4	32.5	1.3184	6000
CBO	7.5	67.5	10	10	1.4	30	1.2927	6000
DPSO	7.5	67.5	10	10	1.4	30	1.2927	6000

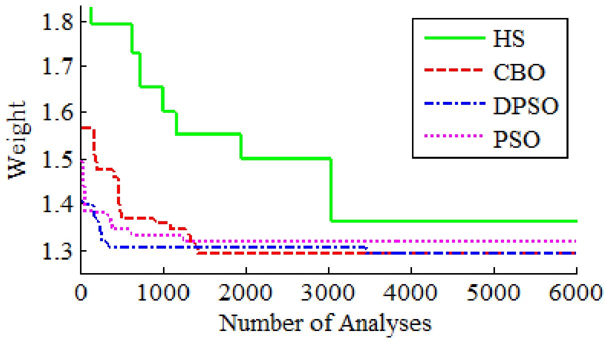

Figure 4. Convergence curves of the HS, DPSO, CBO and PSO algorithms

5. CONCLUDING REMARKS

In this paper, two recently developed meta-heuristic algorithms, known as the Colliding Bodies Optimization and Democratic Particle Swarm Optimization, and one of the most widely used multi-agent meta-heuristic algorithm, known as Particle Swarm Optimization are utilized for optimal design of a concrete ribbed slab. PSO method performs easily in engineering problems, needs little number of parameters and has high power in finding suboptimal solutions in a reasonable amount of time. But one of the PSO's main defects is the problem of premature convergence.

In the standard PSO the next position of a particle is obtained only based on that particle's own experience and that of the best particle ever. On the other hand in the democratic PSO the next position of a particle is decided on based on the attitude of a bigger set of particles called eligible particles. This lets the particles get their information from a more varied set of resources. Moreover, letting particles with seeming lower values of objective function take part in decision making enables the algorithm to represent better exploration characteristics.

CBO utilizes simple formulation to find minimum values of functions and need no internal parameter to be adjusted.

A minimizing problem of a one-way reinforced concrete ribbed slab simply supported at both ends is considered in this paper in order to examine the effectiveness of the above mentioned methods.

The main objective of this paper is to study the convergence curves of these two methods for a concrete ribbed slab and compare the obtained values with results of harmony search method. In this example, the CBO and DPSO performed meaningfully better than the HS because of getting the best solutions so far and CBO attained the best solutions so far in less number of analyses in relation to HS. Although in this example PSO performs better than HS, the democratic PSO achieve better result than the standard PSO by obtaining lower weight and addressing the problem of premature convergence.

The results obtained show that DPSO and CBO methods are powerful and efficient approaches for finding the optimum solution to structural optimization problems. These simple meta-heuristic algorithms can be used in many other engineering design problems to decrease the construction costs.

REFERENCES

1. Kaveh A, Farhoudi N. A unified approach to parameter selection in meta-heuristic algorithms for layout optimization, *Journal of Constructional Steel Research*, **67**(2011) 15453–62.
2. Holland JH. *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.
3. Goldberg DE. *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Boston, USA, 1989.
4. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory, *In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.
5. Gomes HM. Truss optimization with dynamic constraints using a particle swarm algorithm, *Expert Systems with Applications*, **38**(2011) 957–68.
6. Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man and Cybernetics*, **26**(1996) 29–41.
7. Kirkpatrick S, Gelatt C, VecchiM. Optimization by simulated annealing, *Science*, **220**(1983) 671–80.
8. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm; harmony search, *Simulation*, **8**(2001) 60–76.
9. Erol OK, Eksin I. New optimization method: Big Bang–Big Crunch, *Advances in Engineering Software*, **37**(2006) 106–11.
10. Kaveh A, Talatahari S. Size optimization of space trusses using Big Bang–Big Crunch algorithm, *Computers & Structures*, **87**(2009)1129–40.
11. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Machanica*, **213**(2010) 267–86.
12. Kaveh A, Motie Share MA, and Moslehi M. A new meta-heuristic algorithm for optimization: magnetic charged system search, *Acta Mechanica*, No. 1, **224**(2013)85-107.
13. Kaveh A, Talatahari S. Hybrid charged system search and particle swarm optimization for engineering design problems, *Engineering Computations*, No. 4, **28**(2011) 423–40.
14. Fourie PC, Groenwold AA. The particle swarm optimization algorithm in size and shape optimization, *Structural and Multidisciplinary Optimization*, **23**(2002) 259–67.
15. Perez RE, Behdinan K. Particle swarm approach for structural design optimization. *Computers and Structures*, **85**(2007) 1579-88.
16. Jansen PW, Perez RE. Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach, *Computers and Structures*, **89**(2011) 1352-6.
17. Luh GC, Lin CY. Optimal design of truss-structures using particle swarm optimization, *Computers and Structures*, **89**(2011) 2221-32.
18. Zhao X. A perturbed particle swarm algorithm for numerical optimization, *Applied Soft Computing*, **10**(2010) 119–24.
19. Zhao Y, Zub W, Zeng H. A modified particle swarm optimization via particle visual modeling analysis, *Computers & Mathematics with Applications*, **57**(2009) 2022-9.

20. Wang Y, Li B, Weise T, Wang J, Yuan B, Tian Q. Self-adaptive learning based particle swarm optimization, *Information Sciences*, **181**(2011) 4515-38.
21. Kaveh A, Zolghadr A. Democratic PSO for truss layout and size optimization with frequency constraints, *Computers & Structures*, **130**(2014) 10-21.
22. Kaveh A, Mahdavai VR. Colliding bodies optimization: A novel meta-heuristic method, *Computers and Structures*, **139** (2014)18-27.
23. Kaveh A, Mahdavai VR. Colliding Bodies Optimization for discrete optimal design of truss structures, *Computers and Structures*, **139**(2014)43-53.
24. ACI Committee 318, *Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary (318R-08)*, American Concrete Institute, Farmington Hills, Michigan, 2008.
25. Gomes MH. Truss optimization with dynamic constraints using a particle swarm algorithm, *Expert Systems with Applications*, **38**(2011) 957-68.
26. Eberhart RC, Shi Y. Particle swarm optimization: Developments, applications, and resources, *Proceedings of 2001 congress on evolutionary computation*, IEEE Press, Piscataway, NJ, 2001, pp. 81-6.
27. Bergh F, Engelbrecht AP. A study of particle swarm optimization particle trajectories, *Information Sciences*, **176**(2006) 937–71.
28. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Machanica*, **213**(2010) 267-89.
29. Kaveh A, Shakouri Mahmud Abadi A. Cost Optimization of Reinforced Concrete One-Way Ribbed Slabs Using Harmony Search Algorithm, *Arabian Journal for Science and Engineering*, **36**(2011) 1179-87.