ASIAN JOURNAL OF CIVIL ENGINEERING (BHRC) VOL. 16, NO. 3 (2015)
PAGES 437-449

OPTIMIZATION OF LATTICED COLUMNS USING DEMOCRATIC PSO AND CBO ALGORITHMS

A. Kaveh* and Sh. Hasana
Center of Excellence for Fundamental Studies in Structural Engineering, Iran University of science and Technology, Narmak, Tehran 16, Iran

Received: 25 August 2014; **Accepted:** 29 November 2014

ABSTRACT

In this paper, optimal design of latticed columns is performed under static loads utilizing two new algorithms, Colliding Bodies Optimization (CBO) and Democratic Particles Swarm Optimization (DPSO) and also a comparison between two valid codes, AISC 360-10 and Eurocode 3, is investigated. This optimization is on the basis of cost function of materials used in latticed columns, according to each standard and their constructions. Three examples are optimized for each code and their convergence curves are compared. Finally a comparison between two codes is done and the most optimum standard is presented.

Keywords: Colliding bodies optimization (CBO); democratic particle swarm optimization (dpso); built-up column; latticed column; optimal design.

1. INTRODUCTION

Nowadays due to increasing the structural dimensions, the weight of the structures are ncreased thus the engineers tend to use members with high strength plus better architectural and economical properties. The total cost of a steel structure consists of the price of the material (30-73)% and the rest of the cost, such as manufacture (16–22)%, assembling (5–20)%, transportation (3–7)%, and design (2–3)%, having minimum contribution on the total cost. Therefore, choosing optimal shape and optimum parameters of consumed profiles, reduce the consumption of the material and its costs [1].

Built up members have special importance in frame structures works particularly when the goal is to maximize the bearing capacity along with minimum final structural weight. Tower crane, truck crane booms, booms, telecommunication tower, latticed and battened columns are some of applications of built up members. Also when we have long buckling length and low compression load, using this type of compression member is useful. Built up

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

columns compared with rolled sections with same cross section are much lighter. Usually rolled sections are used in constructions. Sometimes existing profiles cannot provide the required bearing capacity, then built up members can be considered as a suitable replacement.

Other methods for making the structures lighter is utilize optimum sections. For this purpose different optimization methods are presented. Optimization methods are important in engineering design. Methods of optimization can be divided into two general categories: 1. Mathematical methods such as Newton-Raphson, differentiation approaches, quasi-Newton (QN) and dynamic programming (DP) [2]; 2. Optimization algorithms that can be divided in to two part meta-heuristic algorithms such as Genetic algorithms (GA) [3], Particle swarm optimization (PSO) [4], Ant colony optimization (ACO) [5], Big bang big crunch (BB-BC) [6], Charged system search (CSS) [7], Ray optimization (RO) [8], Democratic PSO [9], Dolphin echolocation (DE) [10], Mine blast (MB) [11], Colliding bodies optimization (CBO) [12,13]. This algotithm in enhanced by Kaveh and Ilchi Gazaan [14,15]. These algorithms usually mimic the phenomena from nature.

In the following sections a latticed column is optimized by two meta-heuristic algorithms, namely the particles swarm optimization and colliding bodies optimization. In section 2, we introduce design variables. In section 3 two utilized optimization algorithms are briefly presented. In section 4 the process of latticed column design via two standard codes, AISC and Euro code is described. In section 5 an example is presented and finally in section 6 conclusions are derived.

2. DESIGN VARIABLES OF THE PROBLEM

Fig. 1 shows the four design variables considered for modeling of the latticed column. These variables consist of two continues ones: the throat thickness of fillet welding (x(1)), distance between two main profiles (x(2)) and two discrete ones: number of I shaped profile as main profile (x(3)), number of angle profile as lacing (x(4)). These two discrete variables are selected from a list of sections properties according to each standard and in fact they are counter. It should be noted that the distance between two successive lacings, can also be a variable, however, it is related to the variable (x(2)) by the following formula:

lacing's length=x(2)/sin (alpha)

a=2*(lacing's length *cos(alpha)+Lacing's flange length/(2*sin(alpha))+x(1));

3. OPTIMIZATION ALGORITHMS

In this section three optimization algorithms is presented. The first one is Democratic PSO, that is introduced as an improved version of particle swarm optimization algorithm and the second one is a new meta-heuristic algorithm so-called colliding bodies optimization (CBO) and in the follow enhanced version of CBO (ECBO) is introduced.

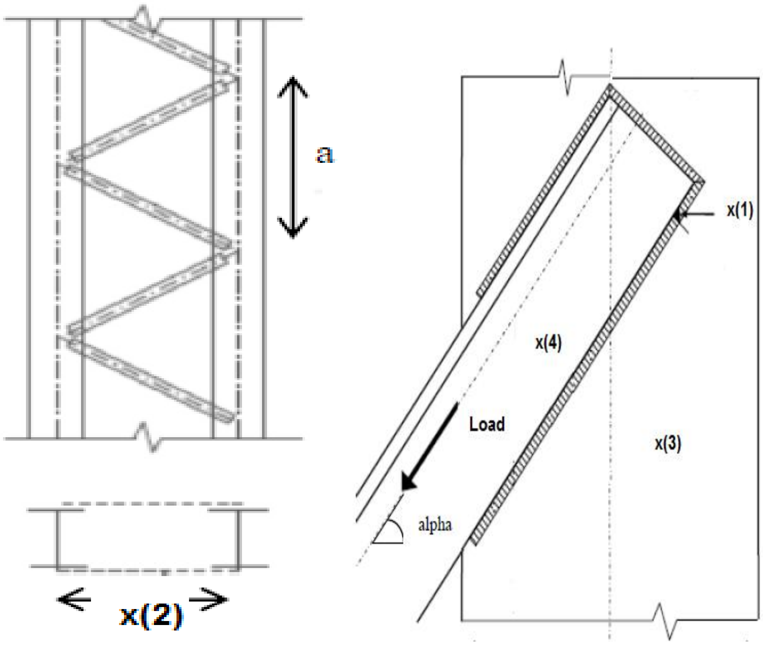

Figure. 1 Details of a latticed column design

3.1 Democratic particle swarm optimization

Democratic PSO optimization algorithm is an improved version of Particle Swarm Optimization (PSO) optimization algorithm, developed by Kaveh and Zolghadr [9]. PSO was first introduced by Kennedy and Eberhart [4], is a population-base meta-heuristic algorithm inspired by the social behavior of animals such as fishes schooling, insects swarming, and birds flocking. Like any other population-based meta-heuristic algorithm, PSO starts with a set of agents which are randomly spread in the multi-dimensional search space of problem. As the optimization process continues these agents move around in the search space searching for better positions. By gradual improvement of the locations of the particles in a swarm the algorithm finally converges to a sub-optimal solution. However, PSO is notorious for premature convergence. In fact, it lacks proper exploration capability. In the standard PSO all the particles are just being eagerly attracted toward better solutions. And each particle, moving toward the best position experienced by itself and by the whole swarm and some better regions of the search space that experienced by other particles being disregarded.

Indeed, the particles of the standard PSO are only motivated by their own preference and the best particle's dictation. Except for their own knowledge and that of the best particle , they do not take the achievements of the other members of the swarm into account i.e. the information is not appropriately shared between the particles of the swarm.

In order to address this problem, the velocity vector of the democratic PSO is defined as:

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

The main different between the standard PSO and the Democratic PSO is in the velocity vector, exactly in the last term of the previous equation. The term $c_3 r_3 d_{i,j}^k$ represents the democratic effect of the other particles of the swarm on the movement of the *i*th particle, where:

c_3 a parameter to control the weight of the democratic vector
r_3 a random number uniformly distributed in the range (1,0)
$d_{i,j}^k$ this term is produce as follow:

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

$$Q_{ik} = \frac{E_{ik} \frac{obj_{\,best}}{obj\,(k)}}{\sum_{k=1}^{n} E_{ik} \frac{obj_{\,best}}{obj\,(k)}}$$

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

where:

Q_{ik} the weight of the kth particle in the democratic movement vector of the ith particle
X the particle's position vector
obj_{best} the value of the objective function for the best particle in the current iteration
obj_{worst} the value of the objective function for the worst particle in the current iteration
$obj(k)$ objective function value for kth particle

Improving the exploration capabilities of the algorithm can be done in two ways:

1) Help the agents to receive information about good regions of the search space other than those experienced by themselves and the best particle of the swarm.

2) Let some bad particles take part in the movement of the swarm.

3.2 Colliding bodies optimization

Colliding Bodies Optimization (CBO) is an efficient and simple continuous optimization algorithm that is developed recently by Kaveh and Mahdavi [12,13]. The main feature of this algorithm is based on collision between objects. In this algorithm each agents are named CBs and the main mass, position and velocity are attached to them. After collision between CBs their velocity and position are updated according to collision lows. In the next part details of this algorithm is presented.

3.2.1 The CBO algorithm

In CBO each solution candidate X_i that itself is a vector containing a number of variables ($X_i=\{X_{i,j}\}$) is considered as a colliding body (CB). The massed objects are composed of two main equal groups, stationary and moving objects, where the moving objects follow stationary ones. Two main goals are prevailing here: first improving the moving CB's position and second to push stationary objects towards better positions. After collision position of colliding bodies are updated based on the collision laws.

The CBO procedure can briefly explained as follow:

1. The algorithm starts with a random initial position of a main number of agents (CBs) by the follow formula:

$$X^0_i = X_{min} + rand(X_{max}-X_{min}) \quad i=1,2,\ldots,n$$

where:

X^0_i the initial value vector of the *i*th CB

X_{min} and X_{max} minimum and maximum allowable values vectors of the variables

Rand a random number in the interval [0,1]

n number of agents that here named CBs and must be even

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

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

Where:

$fit(i)$ the objective function value of the *i*th CB.

3. Then CBs objective function values is arranged in ascending form. The sorted CBs are divided into two equal groups:

- The lower half of the CBs are stationary CBs that have lower objective function value. These CBs are good agents and the initial velocity of them is equal to:

$$v_i=0, \quad i=1, 2, \ldots,\frac{n}{2}$$

- The upper half of the CBs are moving ones. These CBs move toward the lower then agents with upper value of each group will collide together. The velocity of these bodies before collision is equal to:

$$v_i=x_i - x_{i-\frac{n}{2}} \quad i=\frac{n}{2}+1,\ldots,n$$

where:

v_i and X_i velocity and position vector of the ith CB in this group

$X_{i-n/2}$ the ith CB pair position of X_i in the previous group

4. After the collision, the velocity of moving CBs are derived as follow:

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

where:

v_i and v_i' the velocity of *i*th moving CB before and after collision, respectively;

m_i *and* $m_{i-\frac{n}{2}}$ masses of the *i*th CB and the *i*th CB pair, respectively;

Also, the velocity of stationary CBs after the collision is:

$$v_i^{'}=\frac{(m_{i+\frac{n}{2}}+\varepsilon m_{i+\frac{n}{2}})v_{i+\frac{n}{2}}}{m_i+m_{i+\frac{n}{2}}} \quad i=1,\dots,\frac{n}{2}$$

Where:

$v_{i+\frac{n}{2}}$ and $v_i^{'}$ the velocity of *i*th moving CB pair before and *i*th stationary CB after the collision respectively;

m_i and $m_{i+\frac{n}{2}}$ mass of the *i*th CB and mass of the *i*th moving CB pair respectively;

ε the coefficient of restitution (COR) that is derived by the follow equation:

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

5. New position of the CBs are evaluated using their velocities after the collision in position of the stationary CBs.

The new positions of each moving CBs is:

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

where:

x_i^{new} and $v_i^{'}$: the new position and the velocity after the collision of the ith moving CB, respectively.

$x_{i-\frac{n}{2}}$: the old position of ith stationary CB pair

The new positions of stationary CBs is:

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

where:

x_i^{new}, x_i and $v_i^{'}$ the new position, the old position and the velocity after the collision of the *i*th stationary CB, respectively.

$rand$ *a* random vector uniformly distributed in the range (-1,1)

The sign '∘' denotes an element by element multiplication.

The process of CBO algorithm is repeated from step 2 until a termination criterion, such as maximum iteration number, is satisfied. The penalty function approach was used for constraint handling. The $fit(i)$ function corresponds to the effective cost. If optimization constraints are satisfied, there is no penalty; otherwise the value of penalty is calculated as the ratio between the violation and allowable limit.

3.2.2 The ECBO algorithm

Enhanced Colliding Bodies Optimization (ECBO) is an improved version of the CBO that is developed recently by Kaveh and Ilchi Ghazaan [14,15]. The ECBO adds a memory to the CBO, to save a number of historically best CBs and utilizes a mechanism to escape from local optima. The main alteration for exchanging the CBO to the ECBO, is after step 3 of the presented CBO algorithm. In this step, colliding memory (CM) is utilized to save a number of historically best CB vectors and their related mass and objective function values. Solution vectors which are saved in CM are added to the population and the same number of current worst CBs are deleted. Finally, CBs are sorted according to their masses in a decreasing order again.

Also a parameter so-called Pro within (0, 1) is introduced and it is specified whether a component of each CB must be changed or not. This parameter is used at the end of the CBO algorithm. For each colliding body *Pro* is compared with rn_i (i=1,2,…,n) which is a random number uniformly distributed within (0, 1). If rn_i < pro, one dimension of the *i*th CB is selected randomly and its value is regenerated as follows:

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

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

3.3 Numerical design example

As explained before, using a latticed column, where we have long buckling length or low buckling load, is an economical column. Also we described about optimization approaches and importance of optimization in constructions. Here we combine these two subjects. In this section a latticed column which is designed by two AISC and Euro code standard, is optimized via four algorithms, CBO, ECBO, PSO and Democratic PSO. In this section objective function of design optimization of latticed column is evaluated. The objective function here is the built up column's weight that should be minimized. The column is composed of two main profile with I shaped section, angle profile used as lattice with a certain length and length of welding connection. Then weight of the sum of these parts produce the objective function. Details of evaluating objective function is considered Table 1.

Table 1: Variables, used symbols and their limits in both standard AISC and Eurocode

Variables	AISC (Ib-in)		Euro code (kN-mm)	
	Symbol	limits	Symbol	limits
X(1): throat thickness	a_w	0.125-1.625	a_w	3-35
X(2): distance between two main profile	b	3.94-31.4961	h_0	100-800
X(3): main profile number of section list	i	1-274	i	1-24
X(4): lacing profile number of section list	j	1-127	j	1-107

It should be noted that variable x(2) must be at least equal to the main profile flange. Then we have to apply some limitation on this variable in objective function. For upper limit of this variable we use of the concept that a double column is optimum when the minimum gyration ratio is equal to maximum gyration ratio. For this purpose we use a relationship as follow:

AISC: Euro code:

b=$max(bf$, $|x(2)|$) h_o=max(bf , $|x(2)|$)
b=min(31.4961 , $|x(2)|$) h_o=min(800,$|x(2)|$)

Common symbols in AISC and Euro code:

chord_weight: Weight per unit length of used main profile (I shaped profile),
lacing_weight: Weight per unit length of used lacing (angle profile),
welding_length: Total length of the welding.
L Length of the built up column;
a Length of the main profile between two successive lacings according to Fig. 1
tf Thickness of chord's flange
bf Length of main profile's flange
bfo Length of the lacing profile's flange
tfo Thickness of the lacing's flange

Euro code symbols:

d Length of lacing
N_{ed} Applied load
M_{ed1} Applied moment
N_{ched} Maximum design force for two identical chords
N_{bzrd} The design buckling resistance of the chord about the weak axis of the cross-section,
calculated according to EN 1993-1-1 § 6.3.1
N_{byrd} The design buckling resistance of the chord about the strong axis of the cross-section,
calculated according to EN 1993-1-1 § 6.3.1.
N_{ded} The compression axial force in a diagonal
N_{bdrd} The design buckling resistance of diagonal
N_{ted} Maximum design value of the tensile axial force of diagonal
N_{trd} The design tension resistance
F_{wed} The design value of the force per unit length
F_{wrd} The design weld resistance per unit length

AISC symbols:

P Applied load
P_n Nominal compressive strength
k_i Effective length ratio according to AISC360-10 E6-2 is equal to 0.86
r_{yi} Radius of gyration about y axis of a main profile
slenderness Slenderness ratio of built up column about no material axis
slenderness_lacing Maximum slenderness ratio of lacing
if welding_length $\leq 100a_w$
l_e = welding_length
elseif welding_length $\geq 300a_w$
$l_e = 180a_w$
else
B = min(1.2-0.002×(welding_length/aw),1);
l_e=B×welding_length;
over_lap = bf/(2*sind(alpha))-bfo/2*cotd(alpha)

steel's special gravity= $\mathbf{7850 \times 10^{-9} kg/mm^3 \ = 28360.4172 \times 10^{-5} \ Ib/in^3}$

3.3.1.1 Objective Function

Euro code:

$$\mathbf{2\ chord_{weight} \times L \times 10^{-3} + 2 \times \left[\frac{2L}{a}\right] \times lacing_{weight} \times d \times 10^{-3} + 4 \times welding_{length} \times \frac{{a_w}^2}{2} \times 7850 \times 10^{-9}}$$

AISC:

$$\mathbf{2\ chord_{weight} \times \frac{L}{12} + 2 \times \left[\frac{2L}{a}\right] \times lacing_{weight} \times \frac{d}{12} + 4 \times welding_{weight} \times \frac{{a_w}^2}{2} \times 28360.4172 \times 10^{-5}}$$

3.3.1.2 Constraints

Euro code:

<<<<<<<< checking for buckling resistance of a main profile>>>>>>

g_1= N_{ched}-N_{bzrd}<0 according to EN 1993-1-1 § 6.4

<<<<<< checking for out of plane buckling resistance of the main profiles (chords) >>>>

g_2=N_{ched}-N_{byrd}<0

<<<<<<<< checking for buckling resistance of a diagonal>>>>>>

g_3=N_{ded}-N_{bdrd}<0

<<<<<<<< checking for resistance of the diagonals in tension >>>>>>

g_4=N_{ted}-N_{trd}<0

<<<<<< checking for resistance of the diagonal to main profile welded connection >>>>

g_5=F_{wed}-F_{wrd}<0 according to EN 1993-1-8: 4.5.3.3

g_6=max(30,6a_w)-welding_length<0 according to EN 1993-1-8: 4.5.1(2)

g_7=0,3-a_w<0

g_8=a_w-tfo<0

constraint=g1+g2+g3+g4+g5+g6+g7+g8;

AISC:

<<<<<<<< checking for buckling resistance of a main profile>>>>>>

g_1=P-0.9P_n according to E6-1

g_2= $k_i a/r_{yi}$ - $\mathbf{\frac{3}{4}}$ slenderness<0 according to E6-2

g_3=tfo-tf<0

g_4=$k_i a/r_{yi}$ – 140<0 according to E6-2

g_5= slenderness_lacing – 140<0 according to E6-2

<<<<<<<< checking for buckling resistance of a diagonal>>>>>>

g_6=P_lacing-0.9Pn_lacing

g7=b-15<0 'b'for single lacing in built up column should be less than 15 in.

g8=bfo-b/2<0

g9=max(0,3 - $\left[\mathbf{\frac{2L}{a}}\right]$) ;

g_{10}=flange_ratio-0.56$\sqrt{\frac{E}{f_y}}$ according to AISC manual Table B4.1

g_{11}=web_ratio $-1.49\sqrt{\frac{E}{f_y}}$ according to AISC manual table B4.1

<<<<<< checking for resistance of the diagonal to main profile welded connection >>>>
g_{12}= P_lacing - $R_n<0$
<<<<<< checking for minimum throat thickness size >>>>
min_thickness = min (tfo,tf)
if min_thickness $\leq \frac{1}{4}$
$g_{13}=\frac{1}{8}-a_w<0$
else if $\frac{1}{4}<$ min_thickness ≤ 0.5
$g_{13}=\frac{3}{16}-a_w<0$
else if $0.5<$min_thickness$\leq \frac{3}{4}$
$g_{13}=\frac{1}{4}-a_w<0$
else
$g_{13}=\frac{5}{16}-a_w<0$
<<<<<< checking for maximum throat thickness size >>>>
if tfo $<\frac{1}{4}$
g_{14}= 0,a_w-tfo <0
else
$g_{14}=a_w-(tfo-(\frac{\mathbf{1}}{\mathbf{16}}))<0$
$g_{15}=4a_w-l_e<0$ AISC-10: M4-2b
g_{16}=over_lap-max(1,5×min_thickness) <0 AISC-10: M4-2b
constraint=$g_1+g_2+g_3+g_4+g_5+g_6+g_7+g_8+g_9+g_{10}+g_{11}+g_{12}+g_{13}+g_{14}+g_{15}+g_{16}$

3.3.1.3 Numerical results

The Democratic PSO, PSO, CBO and ECBO algorithms are all coded in Matlab software. The analysis and design stages and cost function are created in a function file that is called from the optimization code of each algorithm. The data for the considered latticed column designed with two standards AISC [16] and Euro code [17] are provided in Table 2.

Table 2: Design data base

Data base	AISC (lb-in)		Euro code (kN-mm)	
	variable	magnitude	variable	magnitude
Applied load	P	202328.05	N_{ed}	900
Applied moment	-	-	M_{ed}	0
Built up column length	L	393.7	L	10,000
Modulus of elasticity	E	29×10^6	E	2.1×10^5
Yield stress	F_y	50000	F_y	335
Angle between lacing and main profile as shown in Fig. 1	alpha	60°	alpha	60°

Finally, Compression of the results for four optimization algorithms, is shown in Fig. 2 and Fig. 3. The results of the optimization algorithms and equivalent item of discrete variables that is used in objective function, are presented in Table 3.

Table 3: Optimum results for both standard AISC and Euro code

Design standard	Optimization algorithm	X(1)	X(2)	X(3)	X(4)	Best cost (kg)
AISC	CBO	-0.25799 0.125	11.96947 11.96947	17.58564 W5X19	-72.9459 L5X5X3/8	1066.897
	ECBO	0.25 0.25	11.63367 11.63367	6.088499 W4X13	37.28917 L4X3X5/16	716.0744
	PSO	0.261384 0.261384	8.090498 8.090498	14.59163 W6X16	28.32161 **L2-1/2X2-1/2X3/8**	730.5072
	Democratic PSO	0.125 0.125	10.13687 10.13687	7.207099 W8X13	35.97097 L4X4X1/4	661.4469
Euro code	CBO	1.366448 3	162.7109 162.7109	2.056833 HE120A	-9.67318 L40X40X5	490.7803
	ECBO	4.36763 4.36763	-175.231 175.231	2.599402 HE120A	9.877333 L40X40X5	490.7209
	PSO	3 3	186.0298 186.0298	1.083832 HE100A	1.309784 L20X20X3	364.5887
	Democratic PSO	3 3	186.2095 186.2095	1 HE100A	1 L20X20X3	364.6182

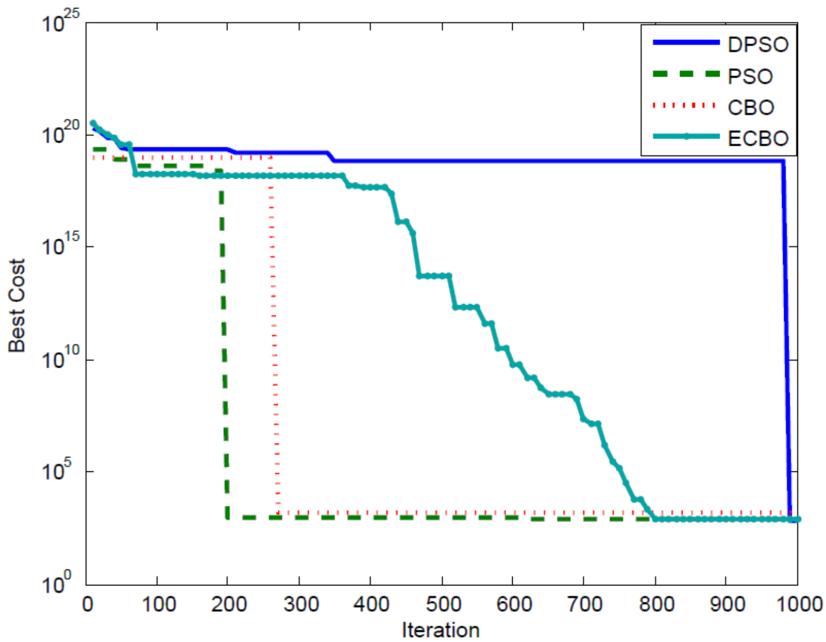

Figure 2. Convergence curves obtained for latticed column designed based on AISC

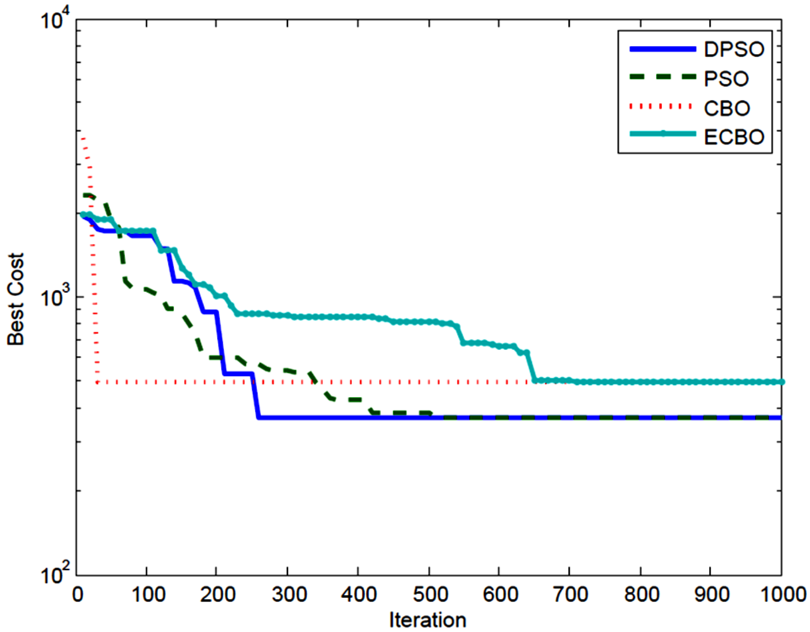

Figure 3. Convergence curves obtained for latticed column designed based on Euro code

3.4 Concluding Remarks

Based on this research the following conclusions can be derived:

According to Table 3, for the objective function based on AISC code, DPSO, ECBO, PSO and CBO lead to the best results, respectively.

According to Table 3, for the objective function based on Euro code, PSO, DPSO, ECBO and CBO lead to the best results, respectively.

As a comparison between two standards, AISC and Euro code, in built up column design, the Euro code standard is more light and economical than AISC; naturally the weights of the used profiles in each standard also effects the weight.

Acknowledgement: The authors are grateful to V.R. Mahdavi for his help in this research.

REFERENCES

1. Farkas J. *Optimum Design of Metal Structures*, Akademiai KIADO, Budapest, 1984.
2. Fogel LJ, Owens AJ, Walsh MJ. *Artificial Intelligence through Simulated Evolution*, Chichester, Wiley, 1966.
3. Holland JH. *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press, 1975.
4. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.
5. Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B:*

Cybernetics, **26**(1996) 29-41.
6. Erol OK, Eksin I. New optimization method: big bang-big crunch, *Advances in Engineering Software*, **37**(2006) 106-11.
7. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mechanica*, **213**(2010) 267-89.
8. Kaveh A, Khayatazad M. A novel meta-heuristic method: ray optimization, *Computers and Structures*, **112-113**(2012) 283-94.
9. Kaveh A, Zolghadr A. Democratic PSO for truss layout and size optimization with frequency constraints, *Computers and Structures*, **130**(2014) 10–21.
10. Kaveh A, Farhoudi N. A new optimization method: dolphin echolocation, *Advances in Engineering Software*, **59**(2013) 53-70.
11. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm for optimization of truss structures with discrete variable, *Computers and Structures*, **102**(2012) 49-63.
12. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method, *Computers and Structures*, **139**(2014) 18-27.
13. Kaveh A, Mahdavi VR. Colliding bodies optimization method for optimum discrete design of truss structures, *Computers and Structures*, **139**(2014) 43-53.
14. 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.
15. Kaveh A, Ilchi Ghazaan M. Computer codes for colliding bodies optimization and its enhanced version, *International Journal of Optimization in Civil Engineering*, No. 3, **4**(2014) 321-39.
16. AISC 360-10. Specification for Structural Steel Buildings, An American National Standard, 2010.
17. Eurocode 3: Design of steel structures, part 1-1, General rules and rules for buildings, CEN European Committee for Standardization, 2005.