Comparative Study on Recent Metaheuristic Algorithms in Design Optimization of Cold-Formed Steel Structures

M.P. Saka, S. Carbas, I. Aydogdu, A. Akin and Z.W. Geem

Abstract Sustainable construction aims at reducing the environmental impact of buildings on human health and natural environment by efficiently using energy, resources and reducing waste and pollution. Building construction has the capacity to make a major contribution to a more sustainable future of our World because this industry is one of the largest contributors to global warming. The use of cold-formed steel framing in construction industry provides sustainable construction which requires less material to carry the same load compare to other materials and reduces amount of waste mimum design algorithms are developed for cold-formed steel frames made of thin-walled sections using the recent metaheuristic techniques. The algorithms considered are firefly, cuckoo search, artificial bee colony with levy flight, biogeography-based optimization and teaching-learning-based optimization algorithms. The design algorithms select the cold-formed thin-walled C-sections listed in AISI-LRFD (American Iron and Steel Institution, Load and Resistance Factor Design) in such a way that the design constraints specified by the code are satisfied and the weight of the steel frame is the minimum. A real size cold-formed steel building is optimized by using each of these algorithms and their performance in attaining the optimum designs is compared.

M.P. Saka
Department of Civil Engineering, University of Bahrain, Isa Town, Bahrain
e-mail: mpsaka@uob.edu.bh

S. Carbas
Civil Engineering Department, Karamanoglu Mehmetbey University, Karaman, Turkey
e-mail: scarbas@kmu.edu.tr

I. Aydogdu
Civil Engineering Department, Akdeniz University, Antalya, Turkey
e-mail: aydogdu@akdeniz.edu.tr

A. Akin
Thomas & Betts Corporation, Meyer Steel Structures, Memphis, TN 38125, USA
e-mail: alperakin@yahoo.com

Z.W. Geem (✉)
Department of Energy IT, Gachon University, Seongnam, South Korea
e-mail: geem@gachon.ac.kr

N.D. Lagaros and M. Papadrakakis (eds.), *Engineering and Applied Sciences Optimization*, Computational Methods in Applied Sciences 38,
DOI 10.1007/978-3-319-18320-6_9

Keywords Structural optimization · Discrete optimization · Cold-formed thin-walled steel frames · Metaheuristic techniques · Swarm intelligence · Firefly algorithm · Cuckoo search algorithm · Artificial bee colony algorithm · Biogeography-based optimization algorithm · Teaching-learning-based optimization algorithm

1 Introduction

Structural optimization aims at producing buildings that can be built by using the least amount of materials. This aim is of prime importance today because of the reason that buildings and construction works have the largest single share in global resource use and pollution emission [1]. World's climate is visible changing and its ecosystem is currently leading towards irreversible damages due to global warming. Carbon dioxide is the primary greenhouse gas emitted through human activities which is blamed for the global warming. Although energy production and transportation are two of the major source of carbon dioxide emissions, the construction industry also play important role in this respect. The importance of sustainable construction becomes even more apparent when one considers the fact that urban population swells by around one million people every week.

Steel is one of the most sustainable materials in the world. Since the early 1990s, the steel industry has reduced its energy use to produce a ton of steel by approximately one third. More than 95 % of the water used in the steel making process is recycled and returned. Every piece of steel used in construction contains recycled content. Further, all steel can be recovered and recycled again and again into new high quality products. Steel structures require less material to carry the same load as concrete or masonry or wood structures. The use of cold-formed steel framing in the construction industry even provides further economy. Furthermore cold-formed steel framing construction reduces the amount of waste generated at a site. This is due to the fact that almost the entire building project is pre-engineered and prepared using modern and efficient technology as framing members and panels in workshops or factories which are then transported and assembled in the site [2]. Cold-formed steel framing refers specifically to members in light-frame building construction that are made entirely of sheet steel formed to various shapes at ambient temperatures. The most common shape for cold-formed steel framing members is a lipped channel section although "Z", "C", "tubular", "hat" and other shapes have been used. Figure 1 shows an example of such construction which is environmentally friendly and has high sound and heat insulation.

Cold-formed members are produced from very thin steel sheets where the thickness varies between 0.4 and 6.4 mm. This thickness is very small compare to the widths of walls of member that they buckle before the stresses reach to yield stress when they are subjected to axial load, shear, bending or bearing. Therefore one of their major design criteria is based on the local buckling of walls of these sections [3, 4]. Furthermore open sections whether hot-rolled or cold-formed in general has relatively small torsional rigidity compare to closed sections. Plane sections do not remain

Fig. 1 Steel building of cold-formed thin-walled open sections

plane and warping distortion takes place when subjected to torsional moments. Large warping deformations cause normal stresses in the cross section in addition to shear stresses. *Vlasov's* theory provides simple way of calculating these stresses [5, 6]. This theory extends the simple bending stress formula to cover the normal stresses that come out due to warping by just adding a similar term to the same formula. This additional term necessitates computation of two new cross sectional properties that are called the sectorial coordinate and warping moment of inertia of the cross-section. Normal stresses develop in thin-walled open section due to warping can be larger than the bending stress depending on the magnitude of the torsional moment section subjected to [7, 8]. It is shown in the literature that warping has substantial effect in the optimum design of steel frames made of thin-walled open sections [9]. In [10] strength and stability problems of mono-symmetrical complex thin-walled open section are studied using Vlasov's theory. Local instability is described according to the theory of thin plates and shells. The analytical solution is compared with the one attained from the finite element model constructed using shell elements. It is stated that the analytical results differ from that of finite element model on the stress distribution. However, the differences between maximum stress values are not so large. Lateral buckling of thin-walled beam under uniformly distributed transverse load, small longitudinal force and two different moments located at its both ends is studied in [11].

Several studies are carried out on the optimum shape design of thin-walled open sections of different shapes in last decade [12, 13]. In [14] cross-sectional design optimization is carried out for cold-formed steel channel and lipped channel columns under axial compression passing through the centroid of the cross-section. The design problem is formulated according to the provisions of AISI (American Iron and Steel Institute) [15, 16]. Flexural, torsional and torsional-flexural buckling of columns and flat-width-to-thickness ratio of web, flange and lip are considered as constraints as they are described in [15]. Micro-genetic algorithm is used in obtaining the solution of design optimization problem. Micro-genetic algorithm uses relatively smaller population size compare to genetic algorithm which results in less computational time. In [17] three optimization methods steepest descent, genetic algorithm and simulated annealing are applied to obtain the optimum shape of cold-formed thin-walled steel columns under AISI provisions. Interesting optimum shapes are obtained by the algorithms developed and performances of optimization methods are compared. This work is extended to cover different cross-sectional geometries and boundary conditions in [18]. The literature review carried out reveals the fact that deterministic as well as stochastic optimization techniques are used to determine the solution of the shape optimization problem of cold-formed thin-walled sections. Furthermore, it is also noticed that most of the research has considered cold-formed single beam with different boundary conditions subjected to axial force, bi-axial bending moment and torsional moment. There are not many works on steel frames made of cold-formed sections. In one of the recent study real-coded genetic algorithm is utilized to develop optimum design algorithm for cold-formed steel portal frames which minimizes its cost [19]. The design variables consist of continuous and discrete variables. The spacing between main frames and pitch of the frame are taken as continuous design variables while the section sizes are to be selected from cold-formed steel section list are treated as discrete design variables. Constraints are implemented from Australian Code of Practice for cold-formed steel.

In this study the optimum design algorithm is developed for cold-formed steel frames made of thin-walled open sections. The design constraints are implemented from AISI-LRFD (American Iron and Steel Institute, Load and Resistance Factor Design, American Institute of Steel Construction) [20, 21]. Design constraints include the displacement limitations, inter-story drift restrictions, effective slenderness ratio, strength requirements for beams and combined axial and bending strength requirements which includes the elastic torsional lateral buckling for beam-columns. Furthermore additional constraints are considered to satisfy practical design requirements. The design algorithm selects the cold-formed sections for the frame members from the cold-formed thin-walled C-sections listed in AISI [22] such that the design constraints are satisfied and the weight of the steel frame is the minimum. Five recent metaheuristic algorithms are employed to determine the optimum solution of the design problem formulated and their performance is compared.

2 Discrete Optimum Design of Cold-Formed Steel Frames to AISI-LRFD

The selection of cold-formed thin-walled C-sections for the members of steel frame is required to be carried out in such a way that the frame with the selected C-sections satisfies the serviceability and strength requirements specified by the code of practice while the economy is observed in the overall or material cost of the frame. When the constraints are implemented from AISI-LRFD [15] in the formulation of the design problem the following discrete programming problem is obtained.

Find a vector of integer values **I** (Eq. 1) representing the sequence numbers of C-sections assigned to ng member groups

$$\mathrm{I}^{\mathrm{T}} = \left[\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_{\mathrm{ng}}\right] \tag{1}$$

to minimize the weight (W) of the frame

$$\text{Minimize} \qquad \mathrm{W} = \sum_{\mathrm{k}=1}^{\mathrm{ng}} \mathrm{m}_{\mathrm{k}} \sum_{\mathrm{i}=1}^{\mathrm{nk}} \mathrm{L}_{\mathrm{i}} \tag{2a}$$

Subject to

- ***Serviceability Constraints***:

$$\frac{\delta_{jl}}{L/Ratio} - 1.0 \leq 0 \quad j = 1, 2, \ldots, nsm, l = 1, 2, \ldots, nlc \tag{2b}$$

$$\frac{\Delta_{jl}^{top}}{H/Ratio} - 1.0 \leq 0, \quad j = 1, 2, \ldots, nj_{top}, l = 1, 2, \ldots, nlc \tag{2c}$$

$$\frac{\Delta_{jl}^{oh}}{h_{sx}/Ratio} - 1.0 \leq 0, \quad j = 1, 2, \ldots, n_{st}, l = 1, 2, \ldots, nlc \tag{2d}$$

where, δ_{jl} is the maximum deflection of jth member under the lth load case, L is the length of member, *nsm* is the total number of members where deflections limitations are to be imposed, *nlc* is the number of load cases, H is the height of the frame, nj_{top} is the number of joints on the top story, Δ_{jl}^{top} is the top story displacement of the jth joint under lth load case, n_{st} is the number of story, *nlc* is the number of load cases and Δ_{jl}^{oh} is the story drift of the jth story under lth load case, h_{sx} is the story height and *Ratio* is limitation ratio for lateral displacements described in ASCE Ad Hoc Committee report [23]. According to this report, the accepted range of drift limits by first-order analysis is $1/750$ to $1/250$ times the building height H with a recommended value of $H/400$. The typical limits on the inter-story drift are $1/500$ to $1/200$ times the story height. $1/400$ is used in this study.

- ***Strength Constraints: Combined Tensile Axial Load and Bending***

It is stated in AISI-LRFD that when a cold-formed members are subject to concurrent bending and tensile axial load, the member shall satisfy the interaction equations given C5.1 of [15] which is repeated below.

$$\frac{M_{ux}}{\phi_b M_{nxt}} + \frac{M_{uy}}{\phi_b M_{nyt}} + \frac{T_u}{\phi_t T_n} \leq 1.0 \tag{2e}$$

$$\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} - \frac{T_u}{\phi_t T_n} \leq 1.0 \tag{2f}$$

where,

M_{ux}, M_{uy}	the required flexural strengths [factored moments] with respect to centroidal axes.
$Ø_b$	for flexural strength [moment resistance] equals 0.90 or 0.95 [21].
M_{nxt}, M_{nyt}	$S_{\text{ft}} F_{\text{y}}$ (where, S_{ft} is the section modulus of full unreduced section relative to extreme tension fiber about appropriate axis and F_{y} is the design yield stress).
T_u	required tensile axial strength [factored tension].
$Ø_t$	0.95 [21].
T_{n}	nominal tensile axial strength [resistance].
M_{nx}, M_{ny}	nominal flexural strengths [moment resistances] about centroidal axes.

- ***Strength Constraints: Combined Compressive Axial Load and Bending***

It is stated in AISI-LRFD that when a cold-formed members are subject to concurrent bending and compressive axial load, the member shall satisfy the interaction equations given in C5.2 of [15] which is repeated below.

For $\frac{P_u}{\phi_c P_n} > 0.15$,

$$\frac{P_u}{\phi_c P_n} + \frac{C_{mx} M_{ux}}{\phi_b M_{nx} \alpha_x} + \frac{C_{my} M_{uy}}{\phi_b M_{ny} \alpha_y} \leq 1.0 \tag{2g}$$

$$\frac{P_u}{\phi_c P_{no}} + \frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \leq 1.0 \tag{2h}$$

For $\frac{P_u}{\phi_c P_n} \leq 0.15$,

$$\frac{P_u}{\phi_c P_n} + \frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \leq 1.0 \tag{2i}$$

where,

P_u	required compressive axial strength [factored compressive force].
$Ø_c$	0.85 [21].
M_{ux}, M_{uy}	the required flexural strengths [factored moments] with respect to centroidal axes of effective section.
$Ø_b$	for flexural strength [moment resistance] equals 0.90 or 0.95 [21]
M_{nx}, M_{ny}	the nominal flexural strengths [moment resistances] about centroidal axes.

and

$$\alpha_x = 1 - \frac{P_u}{P_{E_x}} > 0.0, \quad \alpha_y = 1 - \frac{P_u}{P_{E_y}} > 0.0 \tag{2j}$$

where,

$$P_{E_x} = \frac{\pi^2 E I_x}{(K_x L_x)^2}, \qquad P_{E_y} = \frac{\pi^2 E I_y}{(K_y L_y)^2} \tag{2k}$$

where,

I_x	moment of inertia of full unreduced cross section about x axis.
K_x	effective length factor for buckling about x axis.
L_x	unbraced length for bending about x axis.
I_y	moment of inertia of full unreduced cross section about y axis.
K_y	effective length factor for buckling about y axis.
L_y	unbraced length for bending about y axis.
P_{no}	nominal axial strength [resistance] determined in accordance with Section C4 of AISI [22], with $F_{\text{n}} = F_{\text{y}}$.
$C_{\text{m}x}, C_{\text{m}y}$	coefficients taken as 0.85 or 1.0.

- ***Allowable Slenderness Ratio Constraints***:

The maximum allowable slenderness ratio of cold-formed compression members has been limited to 200.

$$\frac{K_x{}^* L_x}{r_x} \text{ or } \frac{K_y{}^* L_y}{r_y} < 200 \tag{2l}$$

where,

K_x	effective length factor for buckling about x axis
L_x	unbraced length for bending about x axis
K_y	effective length factor for buckling about y axis
L_y	unbraced length for bending about y axis
r_x, r_y	radius of gyration of cross section about x and y axes.

- ***Geometric Constraints***:

Geometric constraints are required to make sure that C-section selected for the columns of two consecutive stories are either equal to each other or the one above storey is smaller than the one in the below storey. Similarly when a beam is connected to flange of a column, the flange width of the beam is less than or equal to the flange width of the column in the connection. Furthermore when a beam is connected to the web of a column, the flange width of the beam is less than or equal to $(D - 2t_b)$ of the column web dimensions in the connections where D and t_b are the depth and the flange thickness of C-section as shown in Fig. 2.

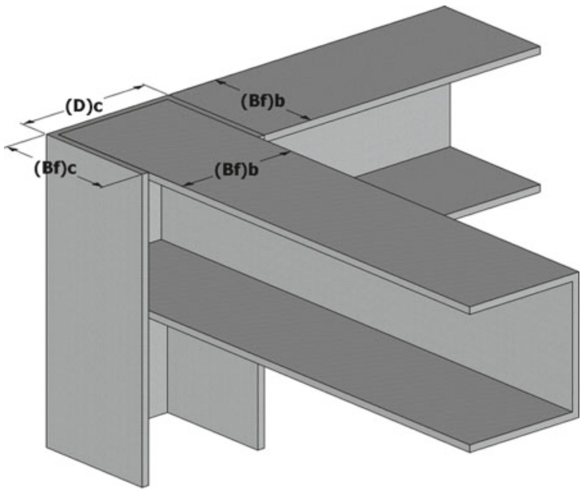


Fig. 2 Typical beam-column connection of C-section

$$\frac{D_i^a}{D_i^b} - 1 \le 0 \quad \text{and} \quad \frac{m_i^a}{m_i^b} - 1 \le 0, \quad i = 1, \ldots, n_{ccj} \tag{2m}$$

$$\frac{B_i^{bi}}{D_i^{ci} - 2t_b^{ci}} - 1 \le 0, \qquad i = 1, \ldots, n_{j1} \tag{2n}$$

$$\frac{B_f^{bi}}{B_f^{ci}} - 1 \le 0, \qquad i = 1, \ldots, n_{j2} \tag{2o}$$

where n_{ccj} is the number of column-to-column geometric constraints defined in the problem, m_i^a is the unit weight of C-section selected for above story, m_i^b is the unit weight of C-section selected for below story, $D_i^a{}_i$ is the depth of C-section selected for above story, D_i^b is the depth of C-section selected for below story, n_{j_1} is the number of joints where beams are connected to the web of a column, n_{j_2} is the number of joints where beams connected to the flange of a column, $D_i^{c_i}$ is the depth of C-section selected for the column at joint i, $t_b^{c_i}$ is the flange thickness of C-section selected for the column at joint i, $B_f^{c_i}$ is the flange width of C-section selected for the column at joint i and $B_f^{b_i}$ is the flange width of C-section selected for the beam at joint i.

Computation of nominal axial tensile strength T_n, nominal axial compressive strength P_n, nominal flexural strengths about centroidal axis M_{nx} and M_{ny} are given in [15] which requires consideration of elastic flexural buckling stress, elastic flexural-torsional buckling stress and distortional buckling strength. Each of these is calculated through use of certain expression given in the design code. Repetition of these expressions is not possible due to lack of space in the article. Hence reader is referred to references [3, 4, 15]. The design problem described through Eqs. 2a–2o turns out to be discrete programming problem. The solution of the design program necessitates selection of cold-formed C-sections from the available list such that the design constraints (2b)–(2o) which are implemented from the design code are satisfied and the objective function given in Eq. 2a has the minimum value.

3 Metaheuristic Algorithms

Obtaining the solution of optimization problems with discrete variables is much harder than solving the optimization problems with continuous variables. Although mathematical programming techniques such as integer programming, branch and bound method and dynamic programming are available for attaining the solution of discrete programming problems, the literature survey related with these techniques reveals the fact they present numerical adversities in finding the solution of large and complex design optimization problems designer face in practice [24, 25]. On the other hand stochastic search methods that are known as metaheuristic techniques are quite efficient in determining the solution of discrete programming problems [26–31]. The fundamental properties of metaheuristic algorithms are that they imitate certain strategies taken from nature, social culture, biology or laws of physics which are used to direct the search process. Their goal is to efficiently explore the search space using these governing mechanisms in order to find near optimal solutions if not global optimum. They also utilize some strategies to avoid getting trapped in confined areas of search space. Furthermore they do not even require an explicit relationship between the objective function and the constraints. They are not problem specific and proven to be very efficient and robust in obtaining the solution of practical engineering design optimization problems with both continuous and discrete design variables [32–34]. In this study the solution of the design optimization problem described in the previous section is obtained by using five recent metaheuristic algorithms and their performance is compared. These are firefly algorithm, cuckoo search algorithm, artificial bee colony algorithm, biogeography-based optimization algorithm and teaching-learning-based optimization algorithms which are developed after 2005. Brief description of each algorithm is given in the following.

3.1 Firefly Algorithm

Firefly algorithm is originated by Yang [35–37] and it is based on the idealized behaviour of flashing characteristics of fireflies. These insects communicate, search for pray and find mates using bioluminescence with varying flaying patterns. The firefly algorithm is based on three rules. These are:

1. All fireflies are unisex so they attract one another.
2. Attractiveness is propositional to firefly brightness. For any couple of flashing fireflies, the less bright one moves towards the brighter one. Attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
3. The brightness of a firefly is affected or determined by the landscape of the objective function.

Attractiveness: In the firefly algorithm attractiveness of a firefly is assumed to be determined by its brightness which is related with the objective function. The brightness i of a firefly at a particular location x can be chosen as $I(x) \propto f(x)$ where $f(x)$ is the objective function. However, the attractiveness β is relative; it should be judged by the other fireflies. Thus, it will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media. In the firefly algorithm the attractiveness function is taken to be proportional to the light intensity by adjacent fireflies and it is defined as;

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad (m \geq 1) \tag{3}$$

where β_0 is the attractiveness at $r = 0$.
Distance: The distance between any two fireflies i and j at x_i and x_j is calculated as

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k} \right)^2} \tag{4}$$

where $x_{i,k}$ is the kth component of the spatial coordinate x_i of the ith firefly.
Movement: The movement of a firefly i which is attracted to another brighter firefly j is determined by

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \left(rand - \frac{1}{2} \right) \tag{5}$$

where the second term is due to the attraction while the third term is randomization with α being the randomization parameter. "*rand*" is a random number generator uniformly distributed in [0, 1].

The values of parameters in the above equations are generally taken as $\beta_0 = 1$ and $\alpha \in [0, 1]$. Randomization term can be extended to a normal distribution N(0, 1) or other distributions. γ characterizes the variation of the attractiveness, and its value determines the speed of convergence and performance of the firefly algorithm. In most applications its value is taken between 0 and 100. The pseudo code of the algorithm is given in [35–37] which is repeated in Fig. 3.

The firefly algorithm is applied to determine engineering as well as structural size, shape and topology design optimization problems [37–39]. In [37] firefly algorithm

Firefly Algorithm

Objective function $f(x)$, $\{x\} = \{x_1, ..., x_d\}^T$

Generate initial population of fireflies $x_i, (i = 1, ..., n)$

Light intensity I_i *at* x_i *is determined by* $f(x_i)$

Define light absorption coefficient γ

***while** (until the termination criteria is satisfied)*

***for i = 1 : n** all **n** fireflies*

***for j = 1 : i** all **n** fireflies*

if $(I_j > I_i)$

Move firefly i towards j in d-dimension

end if

Attractiveness varies with distance r via $\exp[-\gamma r^2]$

Evaluate new solutions and update light intensity

end for j

end for i

Rank the fireflies and find the current best

end while

Postprocess results and visualization

Fig. 3 Pseudo code of firefly algorithm

is used to determine optimum solution of six engineering design problems that are taken from the literature and its performance is compared with other metaheuristic algorithms such as particle swarm optimizer, differential evolution, genetic algorithm, simulated annealing, harmony search method and others. It is stated that the results attained from the optimum solutions of these design examples firefly algorithm is more efficient than particle swarm optimizer, genetic algorithm, simulated annealing and harmony search method.

3.2 Cuckoo Search Algorithm

Cuckoo search algorithm is originated by Yang and Deb [40] which simulates reproduction strategy of cuckoo birds. Some species of cuckoo birds lay their eggs in the nests of other birds so that when the eggs are hatched their chicks are fed by the other birds. Sometimes they even remove existing eggs of host nest in order to give more probability of hatching of their own eggs. Some species of cuckoo birds are even specialized to mimic the pattern and color of the eggs of host birds so that host bird could not recognize their eggs which give more possibility of hatching. In spite of all these efforts to conceal their eggs from the attention of host birds, there is still a possibility that host bird may discover alien eggs. In such cases the host bird either throws these alien eggs away or simply abandons its nest and builds a new one somewhere else. In cuckoo search algorithm cuckoo egg represents a potential solution to the design problem which has a fitness value. The algorithm uses three idealized rules as given in [40]. These are: (a) each cuckoo lays one egg at a time and dumps it in a randomly selected nest. (b) the best nest with high quality eggs will be carried over to the next generation. (c) the number of available host nests is

```
Cuckoo Search Algorithm
Begin;
    Initialize a population of n host nests x_i, i = 1,2,....,n;
    while (until the termination criterion is satisfied);
        Get a cuckoo randomly, (let it be x_i)
            and generate a new solution by Levy flights;
        Evaluate its fitness (let it be F_i);
        Choose a nest among n nests randomly, (let it be x_j);
        if (F_i > F_j)
            replace x_j by the new solution x_i;
        end
        Abandon a fraction (P_a) of worse nests and
        built new ones at new locations via levy flights;
        Keep the best nests (or solutions);
        Rank the solutions and find the current best;
    end while
    Post process results;
end procedure;
```

Fig. 4 Pseudo code for cuckoo search algorithm

fixed and a host bird can discover an alien egg with a probability of $p_a \in [0, 1]$. In this case the host bird can either throw the egg away or abandon the nest to build a completely new one in somewhere else. The pseudo code of the cuckoo search algorithm is given in Fig. 4.

Cuckoo search algorithm initially requires selection of a population of n eggs each of which represents a potential solution to the design problem under consideration. This means that it is necessary to generate n solution vector of $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{ng}\}^{\boldsymbol{T}}$ in a design problem with ng variables. For each potential solution vector the value of objective function $f(\boldsymbol{x})$ is also calculated. The algorithm then generates a new solution $\boldsymbol{x}_i^{\nu+1} = \boldsymbol{x}_i^{\nu} + \beta\boldsymbol{\lambda}$ for cuckoo i where $\boldsymbol{x}_i^{\nu+1}$ and $\boldsymbol{x}_i^{\nu}$ are the previous and new solution vectors. $\beta > 1$ is the step size which is selected according to the design problem under consideration. $\boldsymbol{\lambda}$ is the length of step size which is determined according to random walk with Levy flights. A random walk is a stochastic process in which particles or waves travel along random trajectories consists of taking successive random steps. The search path of a foraging animal can be modeled as random walk. A Levy flight is a random walk in which the steps are defined in terms of the step-lengths which have a certain probability distribution, with the directions of the steps being isotropic and random. Hence Levy flights necessitate selection of a random direction and generation of steps under chosen Levy distribution.

Mantegna [41] algorithm is one of the fast and accurate algorithms which generate a stochastic variable whose probability density is close to Levy stable distribution characterized by arbitrary chosen control parameter $\alpha \quad (0.3 \le \alpha \le 1.99)$. Using the Mantegna algorithm, the step size $\boldsymbol{\lambda}$ is calculated as

$$\lambda = \frac{x}{|y|^{1/\alpha}} \tag{6}$$

where x and y are two normal stochastic variables with standard deviation σ_x and σ_y which are given as

$$\sigma_x(\alpha) = \left[\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\Gamma((1+\alpha)/2)\,\alpha 2^{(\alpha-1)/2}}\right]^{1/\alpha} \quad \text{and} \quad \sigma_y(\alpha) = 1 \text{ for } \alpha = 1.5 \tag{7}$$

in which the capital Greek letter Γ represents the Gamma function ($\Gamma(z) = \int_0^\infty t^{z-1}e^{-z}dt$) that is the extension of the factorial function with its argument shifted down by 1 to real and complex numbers. If z = k is a positive integer $\Gamma(k) = (k-1)!$.

Cuckoo search algorithm is applied to structural optimization problems as well as optimum design of steel frames in [42–44]. It is shown in these applications that cuckoo search algorithm performs better than particle swarm optimizer, big bang-big crunch algorithm and imperialist competitive algorithm. It finds lighter optimum designs.

3.3 Artificial Bee Colony Algorithm with Levy Flight

The artificial bee colony algorithm is suggested by Karaboga et al. [45–50]. It mimics the foraging behaviour of a honey bee colony. In a honey bee colony, there are three types of bees which carry out different tasks. The first group of bees are the *employed bees* that locate food source, evaluate its amount of nectar and keep the location of better sources in their memory. These bees when fly back to hive they share this information to other bees in the dancing area by dancing. The dancing time represents the amount of nectar in the food source. The second group are the *onlooker bees* who observe the dance and may decide to fly to the food source if they find it is worthwhile to visit the food source. Therefore food sources that are rich in the amount of nectar attract more onlooker bees. The third group are *scout bees* that explore new food sources in the vicinity of the hive randomly. The employed bee whose food source has been abandoned by the bees becomes a scout bee. Overall, scout bees carry out the exploration, employed and onlooker bees perform the task of exploitation. Each food source is considered as a possible solution for the optimization problem and the nectar amount of a food source represents the quality of the solution which is identified by its fitness value.

The artificial bee colony algorithm consists of four stages. These stages are initialization phase, employed bees phase, onlooker bees' phase and scout bees phase. These stages are summarized below for the optimization problem of $Min.\ z = f(\mathbf{x})$ where $\mathbf{x}$ is vector of n design variables.

1. ***Initialization phase***: Initialize all the vectors of the population of food sources, $\mathbf{x}_p,\ p = 1, \ldots, np$ by using Eq. 8 where np is the population size (total number of artificial bees). Each food source is a solution vector consisting of n variables ($x_{pi},\ i = 1, \ldots, n$) is a potential solution to the optimization problem.

$$x_{pi} = x_{\ell i} + rand(0, 1)(x_{ui} - x_{\ell i}) \tag{8}$$

where $x_{\ell i}$ and x_{ui} are upper and lower bound on x_i. $rand(0, 1)$ is a random number between 0 and 1.

2. ***Employed bees phase***: Employed bees search new food sources by using Eq. 9.

$$v_{pi} = x_{pi} + \varphi_{pi}(x_{pi} - x_{ki}) \tag{9}$$

where $k \neq i$ is a randomly selected food source, φ_{pi} is a random number in range $[-1, 1]$. After producing the new food source (solution vector) its fitness is calculated. If its fitness is better than x_{pi} the new food source replaces the previous one. The fitness value of the food sources is calculated according to Eq. 10.

$$fitness(x_p) = \frac{1}{1 + f(x_p)} \tag{10}$$

where $f(x_p)$ is the objective function value of food source x_p.

3. ***Onlooker bees' phase***: Unemployed bees consist of two groups. These are onlooker bees and scouts. Employed bees share their food source information with onlooker bees. Onlooker bees choose their food source depending on the probability value P_ℓ which is calculated using the fitness values of each food source in the population as shown in Eq. 11.

$$P_\ell = \frac{fitness(x_p)}{\sum_{p=1}^{np} fitness(x_p)} \tag{11}$$

After a food source x_{pi} for an onlooker bee is probabilistically chosen, a neighbourhood source is determined by using Eq. 8 and its fitness value is computed using Eq. 10.

4. ***Scout bees phase***: The unemloyed bees who choose their food sources randomly called scouts. Employed bees whose solutions cannot be improved after predetermined number of trials (PNT) become scouts and their solutions are abondoned. These scouts start to search for new solutions.

The pseudo code of the artfical bee colony algorithm is given in Fig. 5.

Artificial bee colony algorithm is widely used to obtain the solutions of structural optimization problems [51–57]. It is concluded in these studies that artificial bee colony algorithm is robust and efficient technique that performs better than some other metaheuristic algorithms such as genetic algorithm, ant colony algorithm, particle swarm optimizer, big bang-big crunch and imperialist competitive algorithms.

Artificial Bee colony Algorithm

Initialize the population of solutions x_{ij} *and evaluate the population*

while *(until termination criteria is satisfied)*

- *Produce new solutions* v_{ij} *in the neighbourhood of* x_{ij} *for the employed bees using (9)*
- *Apply the greedy selection process between* x_i *and* v_i
- *Calculate the probability values* P_i *for the solutions* x_i *using (11)*
- *Normalize* P_i *values into [0,1]*
- *Produce the new solutions* v_i *for the onlookers from solutions* x_i *selected depending on* P_i *and evaluate their fitness*
- *Apply the greedy selection process for the onlookers between* x_i *and* v_i
- *Determine the abandoned solution, if exists, and replace it with a new randomly produced solution* x_i *for the scout using (8)*
- *Memorize the best food source position (solution) achieved so far*

end while

Fig. 5 Pseudo code of the artificial bee colony algorithm

3.4 Biogeography-Based Optimization Algorithm

Biogeography-based optimization algorithm is developed by Simon [58] which is based on the theory of island biogeography. Mathematical model of biogeography describes the migration and extinction of species between islands. An island is any area of suitable habitat which is isolated from the other habitats. Islands that are friendly to life are said to have high habitat suitability index (HIS). Features that correlate with HSI include such factors as rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature. The variables that characterize habitability are called suitability index variables (SIV). SIVs can be considered the independent variables of the habitat, and HSI can be considered the dependent variable. Naturally habitats with a high HIS tend to have a large number of species while those with a low HSI have a small number of species. Habitats with a high HSI have many species that emigrate to nearby habitats, simply by virtue of the large number of species that they host. Habitats with a high HSI have a low species immigration rate because they are already nearly saturated with species. Therefore, high HSI habitats are more static in their species distribution than low HSI habitats. This fact is used in biogeography based optimization for carrying out migration. Relationship between species count, immigration rate, and emigration rate is shown in Fig. 6 [58], where I refers to the maximum immigration rate, E is the maximum emigration rate, S_0 is the equilibrium number of species and S_{max} is the maximum species count.

The decision to modify each solution is taken based on the immigration rate of the solution. λ_k is the immigration probability of independent variable x_k. If an independent variable is to be replaced, then the emigrating candidate solution is chosen with a probability that is proportional to the emigration probability μ_k which is usually performed using roulette wheel selection.

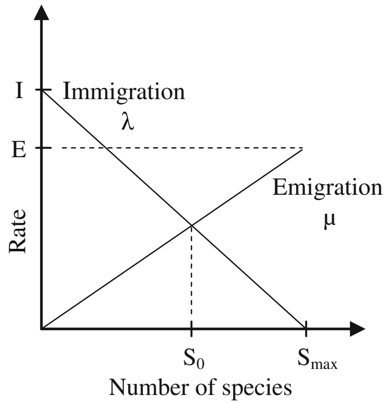


Fig. 6 Species model of a single habitat where λ is immigration rate and μ is emigration rate

$$P(x_j) = \frac{\mu_j}{\sum_{i=1}^{N} \mu_i} \quad \text{for} \quad j = 1, \ldots, N \tag{12}$$

where N is the number of candidate solutions in the population.

Mutation is also another factor which is used to increase the species richness of islands. This increases the diversity among the population. Each candidate solution is associated with a mutation probability defined by

$$m(s) = m_{\max} \left(\frac{1 - P_s}{P_{\max}} \right) \tag{13}$$

m_{max} is a user defined parameter. P_s is the species count of the habitat, P_{max} is the maximum species count. Mutation is carried out on the mutation probability of each habitat. The steps of the biogeography based optimization algorithm can be listed as follows [59].

1. Set up initial population; define the migration and mutation probabilities.
2. Calculate the immigration and emigration rates for each candidate solution in the population
3. Select the island to be modified based on the immigration rate.
4. Using roulette wheel selection on the emigration rate, select the island from which the SIV is to be immigrated.
5. Randomly select an SIV from the island to be emigrated.
6. Perform mutation based on the mutation probability of each island.
7. Calculate the fitness of each individual island
8. If the fitness criterion is satisfied go to step 2.

The pseudo code of biogeography-based optimization algorithm is given in Fig. 7 [60].

Biogeograpy-Based Optimization Algorithm

For each solution y_k, $k \in \{1,....,N\}$, define emigration probability $\mu_k \propto$ fitness of $y_k, \mu_k \in [0,1]$
For each solution y_k define immigration probability $\lambda_k = 1 - \mu_k$
$z \leftarrow y$
For each solution z_k
    For each solution feature s
        Use λ_k to probabilistically decide whether to immigrate to z_k
        If immigrating then
            Use $\{\mu_i\}$ to probabilistically select the emigrating solution y_j
        $z_k(s) \leftarrow y_j(s)$
        end if
    next solution feature
    Probabilistically mutate z_k
next solution
$y \leftarrow z$

Fig. 7 Pseudo code for one generation of biogeography-based optimization algorithm

3.5 Teaching-Learning-Based Optimization Algorithm

Teaching-learning-based optimization algorithm is also population based process which mimics the influence of a teacher on learners [61]. The population represents class of learners. Different design variables in an optimum design problem are considered as different subjects offered to the learners. Learners' achievement is analogous to the fitness value of the objective function. In the entire population the best solution is considered as the teacher. The algorithm consists of two phases; teacher phase and learner phase. In the teacher phase class learns from a teacher and in the learner phase learning takes place through the interaction among the learners.

In the teacher phase the learning process of learners through a teacher is replicated. A good teacher puts an effort to bring the level of learners higher in terms of knowledge. However, in reality it is not only the effort of a teacher which can raise the level of knowledge of learners. The capability of learners also plays an important role in this process. Hence it is a random process. Supposing there are "m" number of subjects (design variables) offered to "n" number of learners (population size, k = 1, 2,…, n). At any sequential teaching-learning cycle i, let T_i be the teacher and M_i be the mean of learners' achievements. T_i will try to move mean M_i to a higher level. After the teaching of T_i there will be a new mean, say M_{new}. The solution is updated according to the difference between the existing and the new mean as:

$$Difference\_Mean = r_i\,(M_{new} - T_F M_i) \tag{14}$$

where T_F is a teaching factor that decides the value of mean to be changed, r_i is a random number in the range of [0, 1]. The value of T_F can be either 1 or 2. It is not a parameter in the algorithm which is computed randomly as T_F = round [1 + rand (0,1) {2-1}]. The difference calculated in Eq. 14 modifies the existing solution as

$$x_{new,i} = x_{old,i} + \ Difference\_Mean \tag{15}$$

In learners' phase the learning process of learners through interaction among themselves is imitated. A learner interacts randomly with other learners with the help of group discussions, presentations, and formal communications. It should be noticed that a learner can learn more unless the other learner has more knowledge than her or him. In this phase randomly two learners say x_i and x_j are selected where $i \neq j$. Learner modification is then expressed as follows:

$$x_{new,i} = x_{old,i} + r_i \left(x_i - x_j\right) \quad \text{if} \quad f(x_i) \ < \ f(x_j) \tag{16}$$

$$x_{new,i} = x_{old,i} + r_i \left(x_j - x_i\right) \quad \text{if} \quad f(x_i) \ > \ f(x_j) \tag{17}$$

$x_{new,i}$ is accepted if it gives a better function value. This process is repeated for the learners in the population. The pseudo code of the algorithm is given in Fig. 8.

Teaching-learning-based optimization algorithm is used to develop structural optimization algorithms in [62–64]. It is shown in these studies that teaching-learning-based optimization algorithm is robust and efficient algorithm that produced better optimum solutions that those metaheuristic algorithms considered for comparison.

Teaching-Learning-Based Optimization Algorithm

```
Initialize the population size and number of generations.
Generate a random population. Calculate the values of objective function for each learner.
  While (number of generation is not reached)
     Calculate the mean of each design variable; x_mean
     Identify the best solution as teacher [ x_teacher ⇒ x with f(x)_min ]
     for i = 1 → n
        Calculate teaching factor T_F,i = round [1 + rand(0,1){2-1}]
        Modify solutions based on teacher  x_new,i = x_i + rand(0,1)[x_teacher − T_F,i x_mean]
        Calculate the objective function value f(x_new,i) for x_new,i
        If f(x_new,i) < f(x_i) then replace x_i = x_new,i
        Select a learner randomly, say x_j such that j ≠ i
        If f(x_i) < f(x_j) then
              x_new,i = x_old,i + r_i(x_i − x_j)
           Else
              x_new,i = x_old,i + r_i(x_j − x_i)
        End if
        If f(x_new,i) < f(x_i) then replace x_i = x_new,i
     End for
  End while
```

Fig. 8 Pseudo code for teaching-learning-based optimization algorithm

4 Constraint Handling

Metaheuristic algorithms are developed to obtain the solution of unconstrained optimization problems. However, almost all of the structural design problems are constrained optimization problems. It is apparent that it becomes necessary to transform the constrained optimum design problem into unconstrained one if one intends to use metaheuristic algorithms for obtaining its solution. One way to achieve this is to utilize a penalty function. In this study the following function is used in this transformation.

$$W_p = W\,(1 + C)^{\varepsilon} \tag{18}$$

where W is the value of objective function of optimum design problem given in 2a. W_p is the penalized weight of structure, C is the value of total constraint violations which is calculated by summing the violation of each individual constraint. ε is penalty coefficient which is taken as 2.0 in this work.

$$C = \sum_{i=1}^{nc} c_i \tag{19}$$

$$c_i = \begin{cases} 0 & if \quad g_j \le 0 \\ g_j & if \quad g_j > 0 \end{cases} \quad j = 1, \ldots, nc \tag{20}$$

where g_j is the jth constraint function and nc is the total number of constraints in the optimum design problem. Constraint functions for the steel frame made of cold-formed sections are given through in Eqs. 2b–2o. It should be reminded that all the constraints are required to be normalized similar to constraint given in Eq. 2n before they are used in the metaheuristic algorithms.

5 Optimum Design Algorithms with Discrete Variables

Five optimum design algorithms are coded each of which is based on the metaheuristic algorithms summarized above. The solution of the discrete optimum design problem given in Eqs. 2a–2o is obtained using these algorithms. In all the optimum design techniques the sequence number of the steel C-sections in the standard list is treated as design variable. For this purpose complete set of 85 C-sections starting from 4CS2x059 to 12CS4x105 as given in AISI [22] is considered as a design pool from which the optimum design algorithms select C-sections for frame members. Once a sequence number is selected, then the sectional designation and properties of that section becomes available from the section table for the algorithm. The metaheuristic algorithms mentioned in Sect. 3 assume continuous design variables. However the design problem considered requires discrete design variables. This necessity is resolved by rounding the numbers obtained through each algorithm. For example Eq. 8 of artificial bee colony algorithm is written as

$$I_{pi} = I_{\min} + INT[rand(0,1)(I_{\max} - I_{\min})], \quad i = 1, \ldots, ng, \quad p = 1, \ldots, np \tag{21}$$

where I_{pi} is the integer value for x_{pi}, the term *rand(0,1)* represents a random number between 0 and 1, I_{min} is equal to 1 and I_{max} is the total number of values in the discrete set for C-section respectively which is equal to 85. *ng* is the total number of design variables and *np* is the number of bees in the colony which is equal to (*neb*+*nob*) where (*neb*) is the number of employed bees and (*nob*) is the number of onlooker bees. The similar adjustments are carried out in other metaheuristic algorithms wherever discrete value are needed for a design variable

The analysis of steel frames is achieved by using matrix displacement method. Noticing the fact that steel frames made of cold-formed thin-walled sections are quite slender structures, large deformations compare to their initial dimensions may take place under external loads. In structures with large displacements, although the material behaves linear elastic, the response of the structure becomes nonlinear [65]. Under certain types of loading, namely, even when small deformations are presumed, nonlinear behavior can be predicted. Changes in stiffness and loads occur as the structure deforms. In such structures, it is necessary to take into account the effect of axial forces to member stiffness. This is achieved by carrying out P-δ analysis in the application of the stiffness method. In each design cycle when the cross sectional properties of members is changed, steel frame is analyzed by constructing the nonlinear stiffness matrix where the interaction between bending moments and axial forces is considered through the use of stability functions. The details of the derivation of the nonlinear stiffness matrix and consideration of geometric nonlinearity in the analysis of steel frames made of thin-walled sections are given in [66].

6 Design Example

Two-storey, 1211-member lightweight cold-formed steel space frame shown in Fig. 9 is selected to study the performance evaluation of five different metaheuristic algorithms. 3-D, plan and floor views of the frame are shown in the same figure respectively. The spacing between columns is decided to be 0.6 m span and each floor has 2.8 m height. The total height of the building is 5.6 m. The space frame consists of 708 joints (including supports) and 1211 members that are grouped into 14 independent member groups which are treated as design variables. The member grouping of the frame is illustrated in Table 1. The frame is subjected to gravity and lateral loads, which are computed as per given in ASCE 7-05 [67]. The loading consists of a design dead load of 2.89 kN/m^2, a design live load of 2.39 kN/m^2, a ground snow load of 0.755 kN/m^2. Unfactored wind load values are taken as 0.6 kN/m^2. The load and combination factors are applied according to code specifications of LRFD-AISC [21] as; Load Case 1: 1.2D+1.6L+0.5S, Load Case 2: 1.2D+0.5L+1.6S and Load

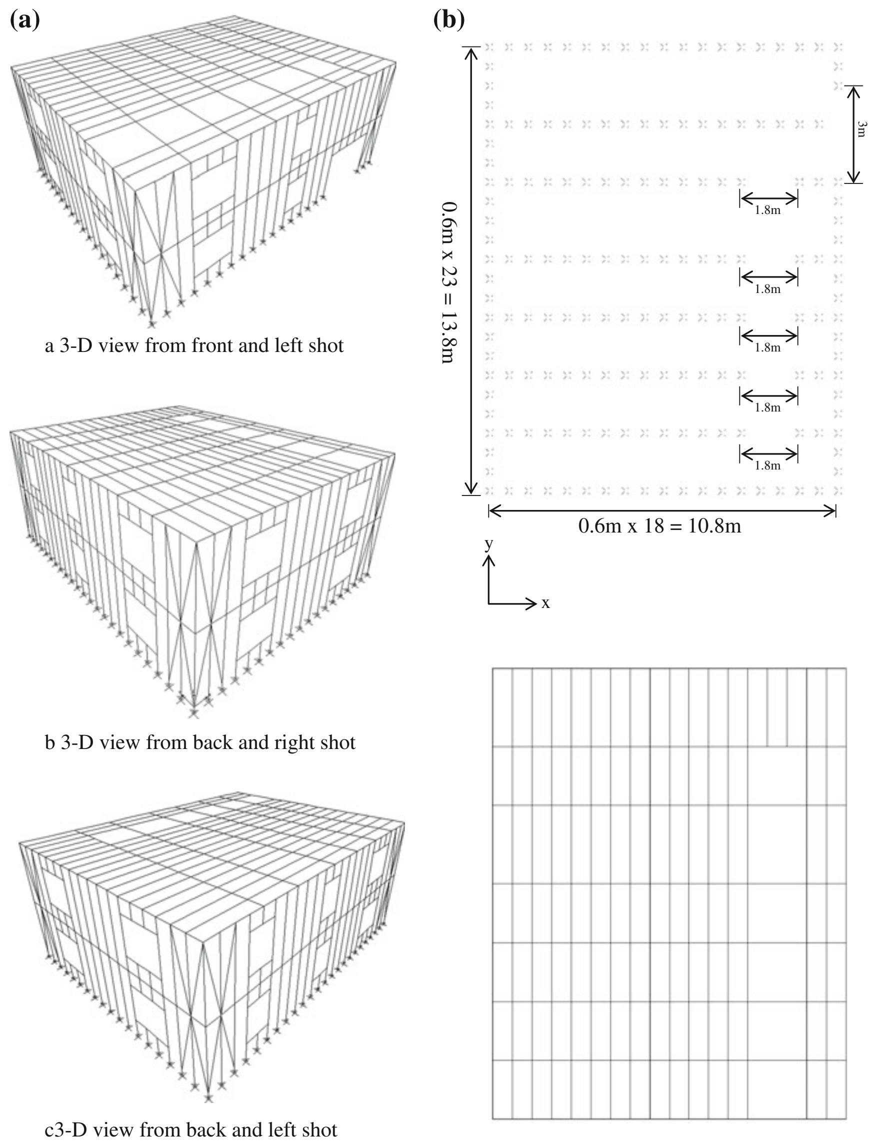
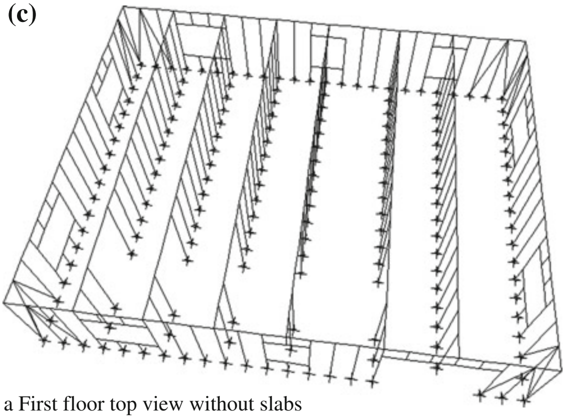


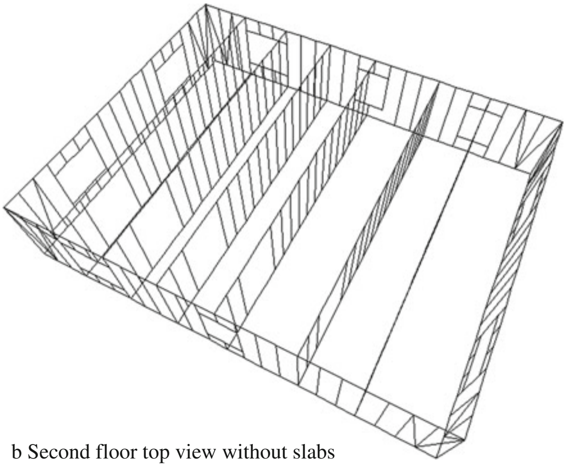
Fig. 9 1211-member three dimensional lightweight cold-formed steel frame, **a** 3-D views from different shots, **b** Plan views, **c** First and second floors *top* views without slabs

Case 3: 1.2D+1.6WX+1.0L+0.5S where D represents dead load, L is live load, S is snow load and WX is the wind load applied on X global direction respectively. The top story drift in both X and Y directions are restricted to 14 mm and inter-story drift limitation is specified to 7 mm. The complete single C-section with lips list

(c)



a First floor top view without slabs



b Second floor top view without slabs

Fig. 9 (continued)

given in AISI Design Manual 2007 [22] which consists of 85 section designations is considered as a design pool for design variables.

The light weight cold-formed steel frame is designed by using five different optimum design algorithm each of which is based on one of the metaheuristic algorithms summarized in Sect. 3. Each metaheuristic algorithm has certain parameters to be initially decided by users. The values adopted for these parameters are given in Table 2 related to each metaheuristic algorithm. Maximum number of iterations is taken as 20,000 for all the algorithms to provide equal opportunity for these techniques. The optimum solutions are obtained after the number of iterations that is much smaller than 20,000.

The optimum designs determined by these five different optimization algorithms are listed in Table 3. It is interesting to notice that all the algorithms have almost found

Table 1 The member grouping of 1211-member lightweight cold-formed steel frame

Storey	Beams outer short	Beams inner short	Beams inner gates	Beams windows	Beams outer gate
1	1	2	3	4	5
2	1	2	3	4	–
Storey	Columns connected short beams	Columns connected long beams	Columns near inner gates	Columns windows	Braces
1	6	7	8	9	14
2	10	11	12	13	14

Table 2 Algorithm parameter values used in the design example

Firefly Algorithm (FFA)	Number of fireflies = 50, $\alpha = 0.5$, $\gamma = 1$, $\beta_{min} = 0.2$, $\beta = 1.0$
Cuckoo Search Algorithm (CSA)	Number of nests = 40, $p_a = 0.90$
Artificial Bee Colony (ABC)	Total number of bees = 50, Maximum cycle number = 400 Limiting value for number of cycles to abandon food source = 250
Biogeography-Based Optimization (BBO)	Population size = 20, Maximum number of generation = 400, Elitism parameter = 2, Mutation probability = 0.01
Teaching-learning-based Optimization (TLBO)	Number of students = 50, Maximum number of generations = 200

optimum designs that are very close to each others. Among all, the Biogeography-Based Optimization (BBO) has attained the best global optimum design with the minimum weight of 53.584 kN (5464.05 kg). The second best solution is determined by Teaching-Learning-Based Optimization (TLBO) where the optimum weight of the frame is 53.677 kN (5473.53 kg) which is only 0.17 % heavier than the optimum design attained by BBO. In fact the difference between the lightest and the heaviest optimum designs is only 1.2 %. This indicates the fact that all these recent metaheuristic algorithms namely firefly algorithm, cuckoo search algorithm, artificial bee colony algorithm, biogeography-based optimization algorithm and teaching-learning-based optimization algorithm are robust and efficient metaheuristic algorithms that are can be used in confidence in solving structural design optimization problems. Inspection of the constraint values given in Table 3 clearly shows that the strength constraints are dominant in the design optimization problem. Almost in all the algorithms the maximum strength ratio is very close to 1.0 while displacement and inter-story drift constraints are much less than their upper bounds.

The convergence history of each algorithm is shown in Fig. 10. It is apparent from this figure that BBO and TLBO have much better convergence rate than firefly (FFA) and artificial bee colony (ABC) algorithms. Although it exhibits rapid conver-

Table 3 Optimum design results of 1211-member lightweight steel frame

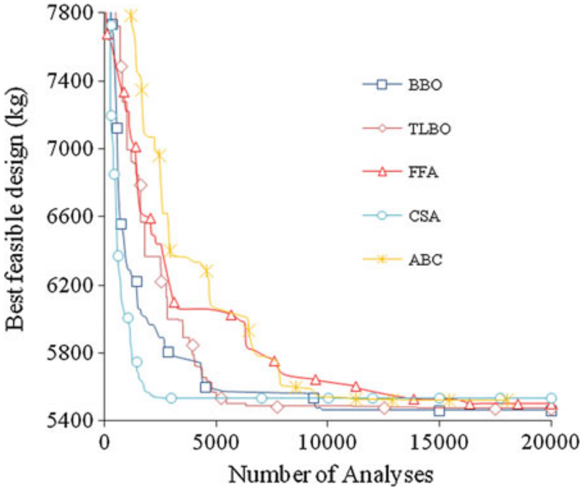
Group No	Group type	Sections selected by BBO algorithm	Sections selected by TLBO algorithm	Sections selected by FFA algorithm	Sections selected by ABC algorithm	Sections selected by CSO algorithm
1	1st and 2nd floors outer short beams	4CS2x085	4CS2x070	4CS2x105	4CS2.5x059	4CS2x070
2	1st and 2nd floors inner short beams	4CS2x059	4CS2x059	4CS2x059	4CS2.5x059	4CS2x059
3	1st and 2nd floors inner gates' beams	8CS2x059	6CS2.5x059	7CS2.5x059	6CS2.5x059	7CS2.5x059
4	1st and 2nd floors windows' beams	4CS2x059	4CS2.5x059	4CS2x059	4CS2x059	4CS2.5x059
5	1st floor outer gate beams	4CS2x059	12CS4x105	4CS2.5x065	4CS2x059	4CS2x059
6	1st floor columns connected short beams	4CS2x059	4CS2x059	4CS2x059	4CS2.5x059	4CS2x059
7	1st floor columns connected long beams	4CS4x059	4CS2x059	4CS2x059	4CS2.5x059	4CS2x059
8	1st floor columns near inner gates	4CS2x059	4CS2x059	4CS2x059	4CS2.5x059	4CS2x059
9	1st floor windows' columns	4CS2x059	4CS4x059	4CS2.5x070	4CS2.5x059	4CS4x059

(continued)

Table 3 (continued)

Group No	Group type	Sections selected by BBO algorithm	Sections selected by TLBO algorithm	Sections selected by FFA algorithm	Sections selected by ABC algorithm	Sections selected by CSO algorithm
10	2nd floor columns connected short beams	4CS2x059	4CS2.5x059	4CS2x059	4CS2x059	4CS2x059
11	2nd floor columns connected long beams	4CS4x059	4CS2x059	4CS2x059	4CS4x059	4CS2x059
12	2nd floor columns near inner gates	12CS3.5x085	12CS4x105	10CS3.5x070	4CS2.5x059	12CS4x105
13	2nd floor windows' columns	4CS2x059	4CS2x059	4CS2.5x070	4CS2x059	4CS2.5x059
14	1st and 2nd floors braces	4CS2x059	4CS2x059	4CS2x059	4CS2x059	4CS2x059
Minimum weight (kN (kg))		53.584 (5464.05)	53.677 (5473.53)	53.962 (5502.61)	54.161 (5522.91)	54.228 (5529.74)
Maximum top storey drift (mm)		9.158	7.885	8.774	7.284	7.568
Maximum inter-storey drift (mm)		1.759	2.433	2.022	2.761	1.882
Maximum deflection (mm)		0.199	0.247	0.198	0.336	0.239
Maximum strength ratio		0.938	0.998	0.937	0.999	0.938
Maximum number of iterations		20,000	20,000	20,000	20,000	20,000

Fig. 10 Search histories of 1211-member lightweight cold-formed steel frame



gence performance to reach optimum solution, the worst design is yielded by CSA producing optimum frame weight as 54.228 kN (5529.74 kg). However considering the fact that the difference between the lightest and heaviest optimum designs is only 1.2 %, it can be concluded that the performance of the all metaheuristic algorithms considered in this study is efficient in this particular design optimization problem.

7 Conclusions

The use of cold-formed thin-walled steel framing in construction industry provides sustainable construction requiring less material to carry the same load. The concept of sustainable building has become quite important due to the rapid increase of human population. The optimum design algorithm developed for cold-formed light weight steel buildings reduces the required amount of material even further level helping the sustainability of the construction. The design procedure selects the optimum cold-formed C-section designations from the section list such that design constraints described in AISI-LRFD are satisfied and the light weight steel frame has the minimum weight. In view of the results obtained it can be concluded that the metaheuristic algorithms considered in this study that are firefly algorithm, cuckoo search algorithm, artificial bee colony optimization algorithm, biogeography-based optimization and teaching-learning-based optimization algorithm all yield an efficient and robust design optimization technique that can successfully be employed in optimum design of light weight cold-formed steel frames. The difference between the heaviest and the lightest optimum designs attained by these algorithms is only 1.2 % which is not significant. The metaheuristic algorithms selected do not require initial selection of too many parameters. Except the firefly algorithm, the rest of the metaheuristic techniques considered needs selection of two parameters, namely population size and maximum number of generations which is the minimum number of parameters that would be required in such procedures. The total number of structural analysis required to reach the optimum design is high similar to most of metaheuristic algorithms. This number may be reduced by carrying out some enhancements in these algorithms such as adding levy flights for random walk. It was not possible to perform comparison of the optimum designs attained in this study with other designs due to the fact that there is no other publication in literature that considers the same design code provisions.

References

1. http://www.isover.com/Our-Commitment-to-sustainability/Toward-sustainable-buildings/. Accessed Oct 2014
2. http://greenmaltese.com/2012/08/cold-formed-steel/. Accessed Oct 2014
3. Ghersi A, Landolfo R, Mazzolani FM (2005) Design of metallic cold-formed thin-walled members. Spon Press, Great Britain
4. Yu W-W, LaBoube RA (2010) Cold-formed steel design, 4th edn. Wiley, New York
5. Vlasov VZ (1961) Thin-walled elastic beams. National Science Foundation, Washington
6. Zbirohowshi-Koscia K (1967) Thin-walled beams. Crosby-Lockwood Ltd, London
7. Trahair NS, Bild S (1990) Elastic biaxial bending and torsion of thin-walled members. Thin-Walled Struct 9:269–307
8. Trahair NS (2003) Lateral buckling strengths of steel angle sections beams. J Struct Eng ASCE 129(6):784–791
9. Aydogdu I, Saka MP (2012) Ant colony optimization of irregular steel frames including elemental warping effect. Adv Eng Softw 44:150–169
10. Magnucki K, Szyc W, Stasiewicz P (2004) Stress state and elastic buckling of a thin-walled beam with mono-symmetrical open cross-section. Thin-Walled Struct 42(1):25–38
11. Magnucka-Blandzi E (2009) Critical state of a thin-walled beam under combined load. Appl Math Modell 33:3093–3098
12. Magnucki K, Monczak T (2000) Optimum shape of open cross section of thin-walled beam. Eng Optim 32:335–351
13. Al-Mosawi S, Saka MP (2000) Optimum shape design of cold-formed thin walled steel sections. Adv Eng Softw 31:851–862
14. Lee J, Kim S-M, Park H-S (2006) Optimum design of cold-formed steel columns by using micro genetic algorithms. Thin-Walled Struct 44:952–960
15. AISI (American Iron and Steel Institute) (2001) North American specification for the design of cold-formed steel structural members
16. AISI (American Iron and Steel Institute) (2002) Cold-formed steel design manual
17. Leng J, Guest JK, Schafer BW (2011) Shape optimization of cold-formed steel columns. Thin-Walled Struct 49:1492–1503
18. Moharrami M, Louhghalam A, Tootkaboni M (2014) Optimal folding of cold-formed steel cross sections under compression. Thin-Walled Struct 76:145–156
19. Phan DT, Lim JBP, Sha W, Siew CYM, Tanyimboh TT, Issa H, Mohammed FA (2013) Design optimization of cold-formed steel portal frames taking into account the effect of building topology. Eng Optim 45(4):415–433
20. AISI (American Iron and Steel Institute) S100–07 (2007) North American specification for the design of cold-formed steel structural members
21. AISC (American Institute of Steel Construction) (1991) LRFD, Volume 1, Structural members, Specifications and code, Manual of steel construction

22. AISI (American Iron and Steel Institute) D100–08 (2008) Excerpts-gross section property tables, cold-formed steel design manual. Part I: Dimensions and properties
23. Ad Hoc Committee on Serviceability (1986) Structural serviceability: a critical appraisal and research needs. J Struct Eng ASCE 112(12):2646–2664
24. Saka MP (2003) Optimum design of skeletal structures: a review, Chapter 10. In: Topping BHV (ed) Progress in civil and structural engineering computing. Saxe-Coburg Publications, Stirlingshire, pp 237–284
25. Saka MP (2007) Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review, Chapter 6. In: Topping BHV (ed) Civil engineering computations: tools and techniques. Saxe-Coburgh Publications, Stirlingshire, pp 105–147
26. Yang X-S (2008) Nature-inspired metaheuristic algorithms. Luniver Press, Bristol
27. Yang X-S (2010) Engineering optimization: an introduction with metaheuristic applications. Wiley, New York
28. Lamberti L, Pappalettere C (2011) Metaheuristic design optimization of skeletal structures: a review. Comput Technol Rev 4:1–32
29. Saka MP (2012) Recent developments in metaheuristic algorithms: a review. Comput Technol Rev 5:31–78
30. Saka MP, Geem ZW (2013) Mathematical and metaheuristic applications in design optimization of steel frame structures: an extensive review. Math Probl Eng
31. Saka MP, Dogan E, Aydogdu I (2013) Review and analysis of swarm-intelligence based algorithms, Chapter 2. In: Yang X-S, Cui Z, Xiao R, Gandomi AH, Karamanoglu M (eds) Swarm intelligence and bio-inspired computation, theory and applications. Elsevier, Amsterdam, pp 25–47
32. Hasançebi O, Çarbaş S, Doğan E, Erdal F, Saka MP (2009) Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Comput Struct 87(5–6):284–302
33. Hasançebi O, Çarbaş S, Doğan E, Erdal F, Saka MP (2010) Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Comput Struct 88(17–18):1033–1048
34. Lagaros ND (2014) A general purpose real-world structural design optimization computing platform. Struct Multidiscip Optim 49(6):1047–1066
35. Yang X-S (2009) Firefly algorithms for multimodal optimization, Chapter. In: Watanabe O, Zeugmann T (eds) Stochastic algorithms: foundations and applications. SAGA 2009, Lecture Notes in Computer Science 5792. Springer, Berlin, pp 169–178
36. Yang X-S (2010) Firefly algorithm, Lévy flights and global optimization, Chapter. In: Bramer M et al (eds) Research and development in intelligent systems XXVI. Springer, London, pp 209–218
37. Gandomi AH, Yang X-S, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89:2325–2336
38. Sayadi MK, Remazanian R, Ghaffari-Nasab N (2010) A discrete firefly metaheuristic with local search for make-span minimization in permutation flow shop scheduling problems. Int J Ind Eng Comput 1:1–10
39. Miguel LFF, Lopez RH, Miguel LFF (2013) Multimodal size, shape, and topology optimization of truss structures using the firefly algorithm. Adv Eng Softw 56:23–37
40. Yang X-S, Deb S (2010) Engineering optimization by cuckoo search. Int J Math Model Numer Optim 1(4):330–343
41. Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. Phys Rev 49(5):4677–4683
42. Gandomi AH, Yang X-S, Alavi AH (2011) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
43. Kaveh A, Bakhspoori T (2011) Optimum design of steel frames using cuckoo search algorithm with Levy flights. Struct Des Tall Spec Build 22(13):1023–1036
44. Saka MP, Dogan E (2012) Design optimization of moment resisting steel frames using a cuckoo search algorithm. In: Topping BHV (ed) Proceedings of the eleventh conference on computational structures technology, Paper 71. Dubrovnik, Croatia, 4–7 Sept

45. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department
46. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 39(3):459–471
47. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214:108–132
48. Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. Adv Soft Comput: Found Fuzzy Logic Soft Comput 4529:789–798
49. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697
50. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large scale problems and engineering design optimization. J Intell Manuf 23:1001–1014
51. Hadidi A, Azad SK, Azad SK (2010) Structural optimization using artificial bee colony algorithm. In: Proceedings of 2nd international conference on engineering optimization, 6–9 Sept
52. Sonmez M (2011) Artificial bee colony algorithm for optimization of truss structures. Appl Soft Comput 11(2):2406–2418
53. Sonmez M (2011) Discrete optimum design of truss structures using artificial bee colony algorithm. Struct Multidiscip Optim 43(1):85–97
54. Talathari S, Nouri M, Tadbiri F (2012) Optimization of skeletal structures using artificial bee colony algorithm. Int J Optim Civil Eng 2(4):557–571
55. Garg H (2014) Solving structural engineering design optimization problems using an artificial bee colony algorithm. J Ind Manage Optim 10(3):777–794
56. Yahya M, Saka MP (2014) Optimum construction site layout planning using multiobjective artificial bee colony algorithm with levy flights. Autom Constr 38(3):14–29
57. Aydogdu I, Akin A, Saka MP (2014) Design optimization of real size steel space frames using artificial bee colony algorithm with Levy flight distribution. Steel Compos Struct (under review)
58. Simon D (2008) Biogeography-based optimization. IEEE Trans Evolut Comput 12(6):702–713
59. Ammu PK, Sivakumar KC, Rejimoan R (2013) Biogeography-based optimization-a survey. Int J Electron Comput Sci Eng 2:154–160
60. Simon D, Rarick R, Ergezer M, Du D (2011) Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. Inf Sci 181:1224–1248
61. Rao RV, Savsani VJ, Vakhari DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43:303–315
62. Togan V (2012) Design of planar steel frames using teaching-learning-based optimization. Eng Struct 34:225–232
63. Togan V (2013) Design of pin jointed structures using teaching-learning-based optimization. Struct Eng Mech 47(2):209–225
64. Degertekin O, Hayalioglu S (2013) Sizing truss structures using teaching-learning-based optimization. Comput Struct 119:177–188
65. Majid KI (1972) Nonlinear structures. Butterworth, London
66. Carbas S (2013) Optimum design of low rise steel frames made of cold-formed thin-walled steel sections. Ph.D. dissertation, Engineering Sciences Department, Middle East Technical University, Ankara, Turkey
67. ASCE 7–05 (2005) Minimum design loads for buildings and other structures, American Society of Civil Engineers