Structural Engineering and Mechanics, *Vol. 52, No. 2 (2014) 351-368*
DOI: http://dx.doi.org/10.12989/sem.2014.52.2.351

Solving design optimization problems via hunting search algorithm with Levy flights

Erkan Doğan*

Department of Civil Engineering, Faculty of Engineering, Celal Bayar University, 45160, Manisa, Turkey

(Received February 12, 2014, Revised June 8, 2014, Accepted June 29, 2014)

Abstract. This study presents a hunting search based optimum design algorithm for engineering optimization problems. Hunting search algorithm is an optimum design method inspired by group hunting of animals such as wolves, lions, and dolphins. Each of these hunters employs hunting in a different way. However, they are common in that all of them search for a prey in a group. Hunters encircle the prey and the ring of siege is tightened gradually until it is caught. Hunting search algorithm is employed for the automation of optimum design process, during which the design variables are selected for the minimum objective function value controlled by the design restrictions. Three different examples, namely welded beam, cellular beam and moment resisting steel frame are selected as numerical design problems and solved for the optimum solution. Each example differs in the following ways: Unlike welded beam design problem having continuous design variables, steel frame and cellular beam design problems include discrete design variables. Moreover, while the cellular beam is designed under the provisions of BS 5960, LRFD-AISC (Load and Resistant Factor Design-American Institute of Steel Construction) is considered for the formulation of moment resisting steel frame. Levy Flights is adapted to the simple hunting search algorithm for better search. For comparison, same design examples are also solved by using some other well-known search methods in the literature. Results reveal that hunting search shows good performance in finding optimum solutions for each design problem.

Keywords: structural optimization; cellular beams; steel frames; welded beam design problem; hunting search algorithm; Levy flights

1. Introduction

In last three decades, optimum design of steel structures has been an important issue for the structural designers. Several powerful optimization techniques have been introduced to achieve more efficient structural optimization techniques. In the optimum design of steel structures it is required to select appropriate design variables such that the structure has the minimum cost or weight while the behavior and performance limitations of the code are satisfied. Mathematical programming methods, such as linear programming, non-linear programming, integer programming, use gradient information to search the solution space near an initial starting point. In fact, they converge faster and reach the optimum point with higher accuracy. However, they fail to

*Corresponding author, Ph.D., E-mail: erkan.dogan@cbu.edu.tr

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

satisfy the needs of practicing engineers. One of the reasons for this is the fact that most of the mathematical programming techniques developed are based on the assumption of continuous design variables while in reality most of the structural optimization design variables are discrete in nature. Although some mathematical programming techniques, such as branch and bound method and integer programming do allow design variables having discrete values, numerical applications indicate that they are not very efficient for obtaining the optimum solution of the large-scale design problems (Horst *et al*. 1995, Horst 1995).

In recent years, as an alternative to mathematical programming based techniques, several meta-heuristic or evolutionary algorithms have been developed, which combine transition rules and randomness by imitating natural phenomena, including the physical annealing process (simulated annealing) (Kirkpatrick *et al*. 1983), the musical process of searching for a perfect state of harmony (harmony search) (Lee and Geem 2004), biological evolutionary processes (evolutionary algorithm and genetic algorithms) (Keane 1995 and Goldberg 1989), swarm intelligence (particle swarm, ant colony) (Kennedy and Eberhart 1995, Camp 2004), breeding behavior of cuckoo species (cuckoo search) (Yang and Deb 2010). Main goal of researchers introducing these methods is to deal with shortcomings of traditional mathematical programming techniques in solving optimization problems. What makes these techniques quite robust and simple compared to other classical methods is the fact that they do need neither the gradient information nor the convexity of the objective function and constraint functions. The mechanisms used in search of the optimum solution are not deterministic but stochastic. 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. The common features of these algorithms are that they all employ random number and incorporate a set of parameters that require to be adjusted initially. Their performance differs depending on the problem under consideration and the predefined values of these parameters. Metaheuristic techniques are widely applied in optimum design of steel structures (Hasancebi 2007, 2008, Saka 2009). The review of these applications is presented in (Saka 2009). After the successful applications of early meta-heuristic techniques in structural optimization, number of new meta-heuristic algorithms have been emerged which are even more efficient and robust than the earlier methods.

One of the recent additions to these novel optimization algorithms is the hunting search algorithm (Oftadeh *et al*. 2010), which is inspired by group hunting of animals such as lions, wolves, and dolphins. Hunters involved in the hunting group encircle and catch their prey abiding by the certain strategies. For instance, wolves can hunt animals bigger or faster than themselves by relying on this kind of hunt. One prey is selected and the hunting group gradually moves toward it. The hunters avoid standing in the wind such that the prey senses their smell. This concept is used in the constrained problem to avoid prohibited regions. In structural optimization process, each of the hunters indicates one solution for a particular problem. Similar to animals cooperate to find and catch the prey, the optimum design process seeks to find the optimum solution. Originally, hunting search algorithm produces continuous numbers. Continuous design variables are used for many problems in the literature, such as welded beam design problem. However, they cannot be used in the optimum design problem of steel frames and cellular beams where the steel sections are to be selected from a steel profile list which consists of discrete values. Due to its ease in the implementation of computer code, rounding off method, where the real numbers are rounded off the nearest integer numbers, is used to accomplish the discrete solution.

2. Hunting search optimization with Levy flights

A constrained optimum design problem in general can be expressed as follows

$$\text{Min.} f(x_i) \quad i=1,\ldots..,n \tag{1}$$

Subject to

$$g_j(x_i) \le 0 \quad j=1,\ldots..,m$$

$$x_i \in X, \quad X=\{x_1, x_2,\ldots x_n\} \tag{2}$$

where x_i represents the discrete design variable *i*, which is to be selected from the set X that contains *q* number of values for these variables. n is the total number of design variables. $f(x_i)$ defines the objective function and $g_j(x_i)$ shows the design constraint *j*. m is the total number of these constraints in the design problem.

Hunting search algorithm is one of the recent additions to the meta-heuristic search techniques of combinatorial optimization problems, introduced by Oftadeh *et al*. (2010). This approach is based on the group hunting of animals such as lions, wolves and dolphins. The common part in the way of hunting of these animals is that they all hunt in a group. They encircle the prey and gradually tighten the ring of siege until they catch the prey. Each member of the group corrects its position based on its own position and the position of other members during this action. If a prey escapes from the ring, hunters reorganize the group to siege the prey again. The hunting search algorithm is based on the way as wolves hunt. The procedure involves a number of hunters which represents the hunting group are initialized randomly in the search space of an objective function. Each hunter represents a candidate solution of the optimum design problem. Originally hunting search algorithm produces continuous design variables. However, in addition to continuous variables, discrete design variables are also used in the present study. To be able to use the method for discrete design variables some adjustments are required to be carried out. Firstly the discrete values among which the values of design variables xi are to be selected in set $\{X\}$ are arranged in ascending sequence. The sequence number of these values is then treated as design variable instead of x_i itself. For example in a design set which consists of 272 values, the sequence numbers from 1 to 272 are the main design variables. At any stage of design cycle, once a sequence number is generated by the algorithm, then the real value of the design variable which corresponds to this sequence number is easily taken from the discrete set. The steps of the algorithm are given in the following:

Step 1. Initialize the parameters: Algorithm has eight parameters that require initial values to be assigned. These are hunting group size (number of solution vectors in the hunting group, *HGS*), maximum movement toward the leader (*MML*), hunting group consideration rate (*HGCR*) which varies between 0 and 1, maximum and minimum values of arbitrary distance radius (Ra^{max} and Ra^{min}), convergence rate parameters (α and β) and number of iterations per epoch (*IE*).

Step 2. Initialize the hunting group: Based on the number of hunters (*HGS*), the hunting group matrix is filled with feasible randomly generated solution vectors. The values of objective function are computed for each solution vector and the leader is defined depending on these values.

Step 3. Generate new hunters' positions: New solution vectors $x'=\{x_1', x_2',\ldots., x_n'\}$ are generated by moving toward the leader (the hunter that has the best position in the group) as follows.

$$x_i' = x_i + r\,(MML)\left(x_i^L - x_i\right) \qquad i = 1,\ldots,n \tag{3}$$

The MML is the maximum movement toward the leader, *r* is a uniform random number [0,1] and x_i' is the position value of the leader for the *i*th variable. If the movement of a hunter toward the leader is successful, it stays in its new position. However, if the movement is not successful, i.e., its previous position is better than its new position it comes back to the previous position. This results in two advantages. First, the hunter is not compared with the worst hunter in the group to allow the weak members to search for other solutions. They may find better solutions. Second advantage is that, for prevention from rapid convergence of the group the hunter compares its current position with its previous position; therefore, good positions will not be eliminated.

Step 4. Position correction- cooperation between members: In order to conduct the hunt more efficiently, the cooperation among hunters should be modeled. After moving toward the leader, hunters tend to choose another position in order to conduct the `hunt' more efficiently, i.e., better solutions. Positions of the hunters can be corrected in two ways; real value correction and digital value correction. In real value correction which is considered in the present study, the new hunter's position $x'=\{x_1', x_2',\ldots, x_n'\}$ is generated from *HG*, on the basis of hunting group considerations or position corrections, which is expressed in Eq. (4). For instance, the value of the first design variable for the *jth* hunter $x_1^{j\prime}$ for the new vector can be selected as a real number from the specified $HG(x_i^1, x_i^2, \ldots, x_i^{HGS})$ or corrected using *HGCR* parameter (chosen between 0 and 1).

$$x_i^{j\prime} \leftarrow \begin{cases} x_i^{j\prime} \in \left\{x_i^1, x_i^2, \ldots, x_i^{HGS}\right\} & \text{with probability } HGCR \\ x_i^{j\prime} = x_i^j \pm Ra & \text{with probability (1- } HGCR) \end{cases} \qquad \begin{matrix} i = 1,\ldots, n \\ j = 1,\ldots, HGS \end{matrix} \tag{4}$$

In Eq. (4), HGCR is the probability of choosing one value from the hunting group stored in the HG. It is reported that values of this parameter between 0.1 and 0.4 produces better results. R*a* is referred to as an arbitrary distance radius for the continuous design variable, which can be reduced or fixed during optimization process. Through the former assumption, Ra can be reduced by use of following exponential function.

$$Ra(it) = Ra_{\min}(x_i^{\max} - x_i^{\min}) \exp\left(\frac{\ell n\left(\frac{Ra_{max}}{Ra_{min}}\right) \times it}{itm}\right) \tag{5}$$

Where it represents the iteration number, $x_i^{\min}$ and $x_i^{\max}$ are the maximum and minimum possible values for x_i. $Ra^{\max}$ and $Ra^{\min}$ denote the maximum and minimum of relative search radius of the hunter, respectively and itm is the maximum number of iterations in the optimization process.

In digital value correction, instead of using real values of each variable, the hunters communicate with each other by the digits of each solution variable. For example, the solution variable with the value of 23.4356 has six meaningful digits. For this solution variable, the hunter chooses a value for the first digit (i.e., 2) based on hunting group considerations or position correction. After the quality of the new hunter position is determined by evaluating the objective function, the hunter moves to this new position; otherwise it keeps its previous position.

Step 5. Reorganize the hunting group: In order to prevent being trapped in a local optimum they must reorganize themselves to get another opportunity to find the optimum point. The algorithm does this in two independent conditions. If the difference between the values of the objective function for the leader and the worst hunter in the group becomes smaller than a preset constant ε1 and the termination criterion is not satisfied, then the algorithm reorganizes the hunting group for each hunter. Alternatively, after a certain number of searches the hunters reorganize themselves. The reorganization is carried out as follows: the leader keeps its position and the other hunters randomly choose their position in the design space.

$$x_i' = x_i^L \pm r\,(x_i^{max} - x_i^{min})\;\alpha\,(-\beta\,(EN)) \tag{6}$$

Where, x_i^L is the position value of the leader for the i^{th} variable, r represents the random number between 0 and 1, x_i^{min} and x_i^{max} are the maximum and minimum possible values for x_i, respectively. *EN* counts the number of times that the hunting group has trapped until this step. As the algorithm goes on, the solution gradually converges to the optimum point. Parameters α and β are positive reals values which determine the convergence rate of the algorithm.

Step 6. Generate hunter's new positions using Levy flights. The algorithm generates a new solution for each hunter (Eq. (7)) by means of Mantegna's algorithm (Mantegna 1994) as follows.

$$x_i' = x_i \pm \beta\lambda r\,(x_i - x_i^L) \tag{7}$$

Where, β>1 is the step size which is selected according to the design problem under consideration, r is random number from standard normal distribution and λ is the length of step size which is determined according to random walk with Levy flights, which can be summarized as in the following.

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 (1994) 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 α ($0.3 \le \alpha < 1.99$). Using the Mantegna algorithm, the step size λ is calculated as

$$\lambda = \frac{x}{|y|^{1/\alpha}} \tag{8}$$

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

$$\sigma_x(\alpha) = \left[\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\Gamma((1+\alpha)/2)\alpha 2^{(\alpha-1)/2}}\right]^{1/\alpha} \quad \text{and} \quad \sigma_y(\alpha) = 1 \quad \text{for } \alpha\text{=1.5} \tag{9}$$

in which the capital Greek letter Γ represents the gamma function that is the extension of the factorial function with its argument shifted down by 1 to real and complex numbers. That is if k is a positive integer. $\Gamma(k) = (k-1)!$

Step7. Terminate the process*:* Steps 3-5 are repeated until maximum number of iterations is satisfied.

2.1 Constraint handling

In this study fly-back mechanism is used for handling the design constraints which is proven to be effective in He *et al*. (2004). Once all hunter positions x_i are generated, the objective functions are evaluated for each of these and the constraints in the problem are then computed with these values to find out whether they violate the design constraints. If one or a number of the hunter gives infeasible solutions, these are discarded and new ones are re-generated. If some hunters are slightly infeasible then such hunters are kept in the solution. These hunters having one or more constraints slightly infeasible are utilized in the design process that might provide a new hunter that may be feasible. This is achieved by using larger error values initially for the acceptability of the new design vectors and then reduce this value gradually during the design cycles and uses finally an error value of 0.001 or whatever necessary value that is required to be selected for the permissible error term towards the end of iterations. This adaptive error strategy is found quite effective in handling the design constraints in large design problems.

3. Discrete optimum design problem of steel frames

The design process of moment resisting steel frames necessitates selection of steel profile sections for its columns and beams from a standard steel section tables. This selection should be carried out in such a way that the frame with the selected steel 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 (LRFD-AISC, 1999) in the formulation of the design problem the following discrete programming problem is obtained.

Find a vector of integer values I (Eq. (10)) representing the sequence numbers of steel sections assigned to *ng* member groups.

$$I^T = \left[I_1, I_2, ..., I_{ng}\right] \tag{10}$$

to minimize the weight (W) of the frame

$$W = \sum_{k=1}^{ng} m_k \sum_{i=1}^{nk} L_i \tag{11}$$

where m_k symbolizes the unit weight of a steel section adopted for member group k. n_k is referred to as the number of members and L_i defines the length of member i.

Subject to

(a) Inter storey drift constraints of the multi-storey frame.

$$\left(\delta_j - \delta_{j-1}\right) / h_j \leq \delta_{ju} \qquad j = 1, ..., ns \tag{12}$$

where; δ_j and δ_{j-1} are lateral deflections of two adjacent storey levels. δ_{ju} represents the allowable lateral displacement. h_j is the storey height and ns is the total number of storeys and in the frame.

(b) Displacement restrictions that may be required to include other than inter-storey drift constraints such as deflections in beams and top storey drift.

$$\delta_i \le \delta_{iu} \qquad i=1,\ldots, nd \tag{13}$$

where; *nd* is the total number of restricted displacements in the frame. The horizontal deflection of columns is restricted due to unfactored imposed load and wind loads to height of column/300 in each storey of a building with more than one storey.

(c) Shear capacity check for beam-columns.

$$V_u \le \varphi V_n \tag{14}$$

where; ϕ, V_u and V_n represent resistance factor in shear, required shear strength, and nominal shear strength, respectively.

(d) The local capacity check for beam-columns.

$$\left(\frac{P_u}{\varphi_c P_n}\right)_{il} + \left(\frac{8}{9}\left(\frac{M_{ux}}{\varphi_b M_{nx}}\right)\right)_{il} \le 1.0 \qquad \text{for} \quad \frac{P_u}{\varphi_c P_n} \ge 0.2 \qquad il = 1,\ldots,nl \tag{15}$$

$$\left(\frac{P_u}{2\varphi_c P_n}\right)_{il} + \left(\frac{M_{ux}}{\varphi_b M_{nx}}\right)_{il} \le 1.0 \qquad \text{for} \quad \frac{P_u}{\varphi_c P_n} \le 0.2 \tag{16}$$

where *l* denotes the load case and *nl* represents the number of load cases, M_{nx} and M_{ux} are referred to as nominal flexural strength and applied moment, respectively. Similarly P_n is nominal axial strength, P_u is applied axial load, ϕ_c is resistance factor for columns if the axial force is in compression, ϕ_b is resistance factor in bending. It is apparent that computation of compressive strength $\phi_c P_n$ of a compression member requires its effective length. Effective length factor and corresponding effective length of a compression member in a frame is determined by using Jackson and Moreland monograph given in McGuire (1968).

(e) Serviceability constraints.

The flange width of the beam section at each beam-column connection at joint *j* should be less than or equal to the flange width of column section.

$$B_{jb} \le B_{jc} \qquad j=1,\ldots, nj \tag{17}$$

where n_j represents the total number of joints in the frame.

Depth and the mass per meter of column section at storey joint s+1 at each column-column connection should be less than or equal to depth and mass of the column section at the lower storey joint *s*.

$$D_{s+1} \le D_s \qquad s = 1,\ldots, nu \tag{18}$$

$$m_{s+1} \le m_s \tag{19}$$

where n_u is the total number of column-column connection constraints.

4. Discrete optimum design problem of cellular beams

Cellular beams can be defined as steel sections with circular openings that are built up by cutting a rolled beam web in a half circular pattern along its centerline and re-welding the two

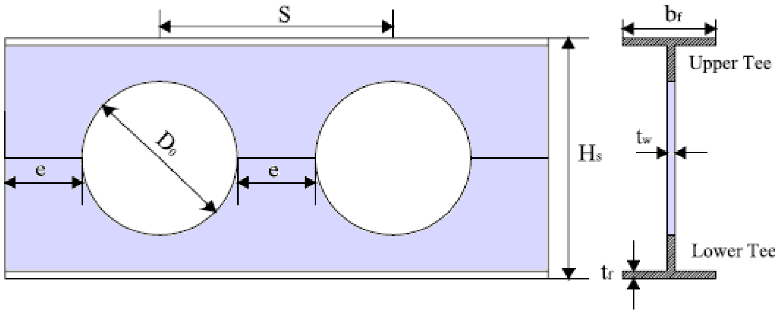

Fig. 1 Geometrical parameters of a cellular beam

halves of hot rolled steel section as shown in Fig. 1. This circular opening results in an increase in the overall beam depth, thereby moment of inertia and section modulus, besides, a decrease in the overall weight of the beam. This consequently leads to approximately 40%-60% deeper and 40%-60% stronger section. Cellular beams, as roof beams beyond the range of portal-frame construction, are generally used at office buildings, parking garages, shopping centers and any structure with a suspended floor and are the perfect solution for curved roof applications, combining weight savings with a low-cost manufacturing process. Cellular beams provide a very economical method of producing tapered members, which have been used extensively in sports stadiums. They can also be used as gable columns and wind-posts.

Optimum design algorithm selects steel UB sections, optimum number of holes and the optimum hole diameter for a cellular beam in such a way that all the design constraints are satisfied and the weight of the beam is minimum. Design provisions are taken from The Steel Construction Institute Publication Number 100 and BS5950. The formulation of the design problem considering these limitations turns out to be a discrete programming problem.

In the design of a cellular beam it is required to the select a UB beam from which the cellular beam is to be produced, to select circular hole diameter and the spacing between the centers of these circular holes or total number of holes in the beam. Therefore, the design variables in the optimum design problem of a cellular beam are selected as the sequence number of a universal beam sections in the standard steel sections tables, the circular hole diameter and the total number of holes. Optimum design algorithm proposed conducts a search from a design pool which consists of list of standard UB beam sections, a list of various diameter sizes and a list of integer numbers starting from 2 to 40 for the total number of holes in a cellular beam. After the consideration of design constraints based on the formulations of cellular beams given in (Erdal *et al*. 2011) in detail, the optimum design problem formulated yields the following mathematical model.

Find a integer design vector **I** representing the design variables

$$\mathbf{I}^{\mathrm{T}} = \left[\mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3 \right] \tag{20}$$

To minimize the weight (W) of the cellular beam

$$\mathrm{W} = \rho \mathrm{A} \mathrm{L} - \rho \left(\pi \left(\frac{\mathrm{D}_0}{2} \right)^2 \mathrm{NH} \right) \tag{21}$$

In Eq.(20), I_1 denotes the sequence number of for the UB beam section in the standard steel sections list, I_2 is the sequence number for the hole diameter in the discrete set which contains various diameter values and I_3 represents the total number of holes for the cellular beam.

In Eq. (21) W is the weight of the cellular beam, ρ denotes the density of steel. A is referred to as the total cross-sectional area of the universal beam section selected for the cellular beam, L represents the span of the cellular beam, D_0 and NH are the diameter of holes and the total number of holes in the cellular beam, respectively.

Following fourteen equations represent the geometrical and behavioral restrictions of cellular beams. These constraints must be satisfied depending on the values of hole diameters, spacing between the hole centers and the final depth of the beam determined.

$$g_1 = 1.08\ D_0 - S \leq 0 \tag{22}$$

$$g_2 = S - 1.60\ D_0 \leq 0 \tag{23}$$

$$g_3 = 1.25\ D_0 - H_S \leq 0 \tag{24}$$

$$g_4 = H_s - 1.75\ D_0 \leq 0 \tag{25}$$

Where; S denotes the distance between centers of holes and H_s is the overall depth of cellular beam.

$$g_5 = M_u - M_p \leq 0 \tag{26}$$

Eq. (26) implies that the maximum moment, M_u, under applied load combinations should not exceed the plastic moment capacity M_p of the cellular beam for a sufficient flexural capacity.

Following three constraints guarantee that the shear stress V_{smax} computed at the supports are smaller than allowable shear stresses P_v, the ones at the web opennings V_{omax} are smaller than allowable vertical shear stresses P_{vy} and finally, the horizontal shear stresses V_{hmax} are smaller than the upper limit P_{vh}.

$$g_6 = V_{smax} - P_V \leq 0 \tag{27}$$

$$g_7 = V_{omax} - P_{vy} \leq 0 \tag{28}$$

$$g_8 = V_{hmax} - P_{vh} \leq 0 \tag{29}$$

It is also required that the web post flexural and buckling capacity of a cellular beam should be at least an allowable value Eq. (30).

$$g_9 = M_W - M_{max} \leq 0 \tag{30}$$

Where M_w is the maximum moment determined at a section of the beam and M_{max} represents the maximum allowable web post moment.

Interaction between the secondary bending stress and the axial force for the critical section in the tee is checked by inequalities (31) and (32).

$$g_{10} = V_T - 0.5\ P_{vy} \leq 0 \tag{31}$$

$$g_{11} = \frac{P_o}{P_u} - \frac{M}{M_p} - 1 \leq 0 \tag{32}$$

Table 1 Parameter set for optimization methods

Optimization method	Parameter set
Hunting Search Algorithm (HuS)	*Number of hunters=40, MML=0.005, HGCR=0.3, Ramax=0.01, Ramin=0.0000001, α=1.2, β=0.02, Maximum number of iterations in one epoch = 25*
Big bang- big crunch (BBBC)	*Number of particles=40, α=1*
Firefly Algorithm (FF)	*Number of fireflies=40, α=1, βmin=0.02, β=1, γ=1*
Cuckoo Search Algorithm (CS)	*Number of nests = 40 ,Pa=0.1*
Particle Swarm Algorithm (PS)	*Number of Particles = 40 ,* c_1*=2,* c_2*=2, w=0.08,* V_{max}*= 2*
Artificial Bee Colony (ABC)	*Total number of bees = 40, Limiting value for number of cycles to abandon food source = 10*

Where; V_T represents the vertical shear on the tee at θ=0 of web opening, P_o and M are the internal forces on the web section. The deflection constraint given in Eq. (33) is imposed such that the maximum displacement is limited to L/360, where y_{max} represents the maximum deflection.

$$g_{12} = y_{max} - \frac{L}{360} \leq 0 \tag{33}$$

5. Performance evaluation of hunting search algorithm with Levy Flights

Above presented hunting search based optimum design algorithms are used to design three different examples. To make a better performance testing of the algorithm, these examples are selected in such a way that they are all different from each other in both geometry and structural behavior. Same parameters given in Table 1 are used for the solution of design problems. Solutions of each design problem obtained with hunting search method are compared with the ones obtained with some other well-known meta-heuristic search techniques in the literature. Each example is solved with each method ten times with different seed values in order to inquire the effect of random numbers to optimum solutions. Algorithms perform this by producing different random number in each iteration by using call random_seed(i) where *i* is the iteration number.

5.1 Welded beam design problem

A carbon steel rectangular cantilever beam design problem is taken from Deb (2000) and selected as first design example. The geometric view and the dimensions of the beam are illustrated in Fig. 2. The beam is designed to carry a certain P load acting at the free tip with minimum overall cost of fabrication. Design variables of the optimization problem can be listed as in the following.

$h= x_1$: the thickness of the weld
$l= x_2$: the length of the welded joint
$t= x_3$: the width of the beam
$b= x_4$: the thickness of the beam

The optimization problem can be stated as follows:

Minimize the cost function;

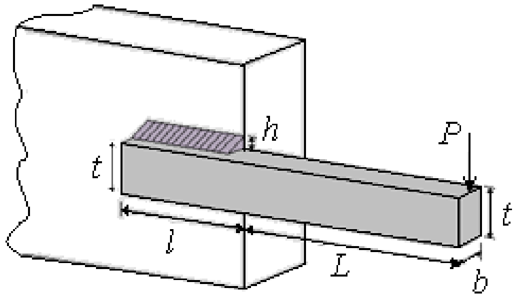

Fig. 2 Welded beam

$$f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

Subject to:

$g_1(x) = \tau(x) - \tau_{max} \le 0$: shear stress

$g_2(x) = \sigma(x) - \sigma_{max} \le 0$: bending stress in the beam

$g_3(x) = x_1 - x_4 \le 0$: side constraint

$g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5 \le 0$: side constraint

$g_5(x) = 0.125 - x_1 \le 0$: side constraint

$g_6(x) = \delta(x) - \delta_{max} \le 0$: end deflection of the beam

$g_7(x) = P - P_c(x) \le 0$: buckling load on the bar

Where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2} x_1 x_2}$$

$$\tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}\, x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\delta(x) = \frac{4PL^3}{Ex_3{}^3 x_4} \quad , \qquad \sigma(x) = \frac{6PL}{x_4 x_3{}^2}$$

$$P_c(x) = \frac{4.013\ E\sqrt{\frac{(x_3^2 x_4^6)}{36}}}{L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}})$$

$P = 6000\ lb, \quad L = 14\ in.,$

$E = 30 \times 10^6\ psi, \quad G = 12 \times 10^6\ psi$

$\tau_{max} = 13{,}600\ psi, \quad \sigma_{max} = 30{,}000\ psi,$

$\delta_{max} = 0.25\ in.$

The side constraints for the design variables are given as follows:

$0.1 \le x_1 \le 2.0, \quad 0.1 \le x_2 \le 10,$

$0.1 \le x_3 \le 10, \quad 0.1 \le x_4 \le 2.0$

Best, worst and average runs obtained by each method are given in Table 2. The optimum designs produced by metaheuristic techniques are tabulated in Table 3. The optimum solution belongs the hunting search algorithm with Levy flights which is 1.724941. This is followed by simple hunting search algorithm with 1.724944, particle swarm algorithm finds the third best optimum design with 1.724956 and artificial bee colony method produces the fourth best value as 1.72808. The design histories of each algorithm are shown in Fig. 3. It is clear from the figure that hunting search with Levy flights has the best convergence rate and the best objective function value obtained is 1.724941.

5.2 Six storey, two bay steel frame

Fig. 4 shows the geometry of a two-bay, six-storey steel frame which is considered as second design example. The frame consists of thirty members that are collected in eight groups as shown

Table 2 Minimum weights (kg) obtained for welded beam

	HuS-L.F	**HuS**	**PS**	**ABC**	**FF**	**BBBC**
Best run	1.724941	1.724944	1.72495	1.72808	1.7312	1.73733
Worst run	1.724941	1.724973	1.72495	1.75141	1.7794	1.73733
Average run	1.724941	1.724955	1.72495	1.73508	1.7371	1,73733

Table 3 Optimum solution for welded beam design problem

Var	Meta-heuristic Search Techniques								
	HuS-L.F	**HuS**	**PS**	**ABC**	**FF**	**BBBC**	**Coell Mont.**	**Rags. Phill.**	**Deb**
$\mathbf{x_1}$	0.205731	0.20573	0.20573	0.20466	0.2015	0.20203	0.2059	0.2455	0.2489
$\mathbf{x_2}$	3.47112	3.47112	3.47107	3.49639	3.5621	3.56690	3.4713	6.1960	6.1730
$\mathbf{x_3}$	9.036624	9.03663	9.03679	9.03407	9.0414	9.03085	9.0202	8.2730	8.1789
$\mathbf{x_4}$	0.205730	0.20573	0.20572	0.20597	0.2057	0.20655	0.2064	0.2455	0.2533
f	**1.724941**	**1.724944**	**1.72495**	**1.72808**	**1.7312**	**1.73733**	**1.7282**	**2.3859**	**2.4331**

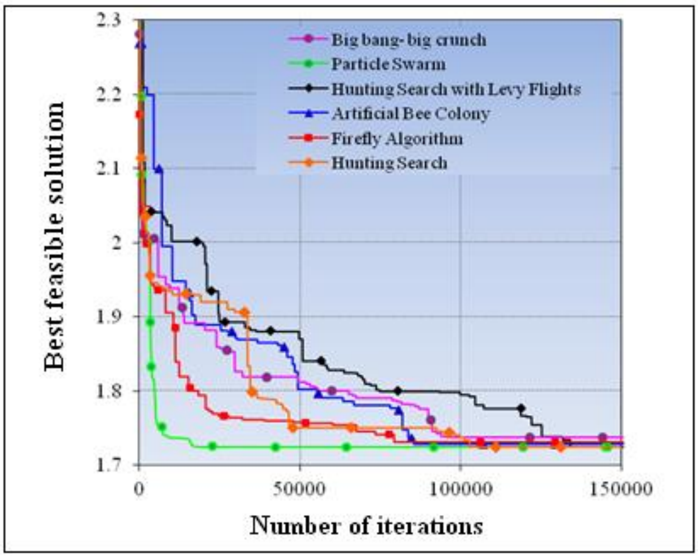

Fig. 3 Design histories of algorithms for welded beam problem

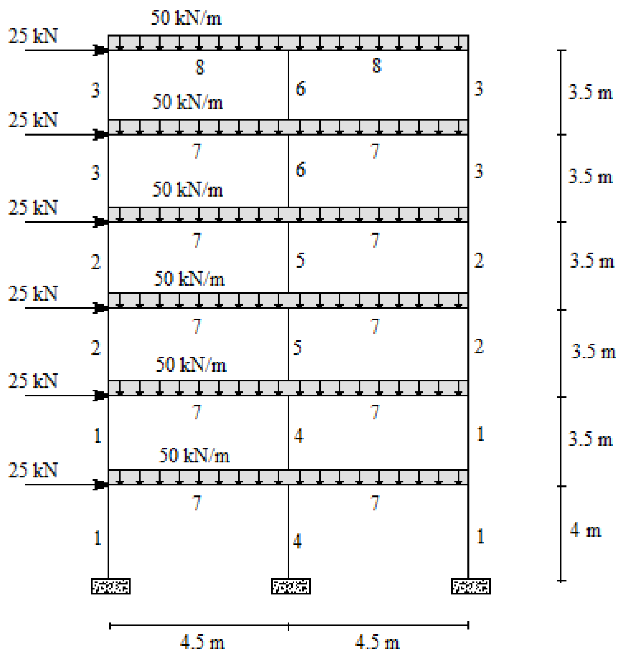

Fig. 4 Six storey two bay steel frame

in the figure. In view of the design restrictions given in LRFD-AISC maximum allowable inter-storey drift is assumed to be 1.17 cm while the lateral displacement of the top storey is limited to 7.17 cm. The modulus of elasticity is 200 kN/mm^2. Complete set of 272 W-sections starting from W100x19.3 to W1100x499mm as given in LRFD-AISC is considered as a design pool from which the optimum design algorithm selects W-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. Consequently the design vector consists of integer numbers from 1 to 272 which corresponds to the sequence numbers of W-sections in the discrete set. Parameter set given in Table 1 is used in the solution of design examples.

Best, worst and average runs obtained by each method are given in Table 4. The optimum W-section designations obtained by the hunting search algorithm with Levy flights are given in Table 5. The minimum weight of the best design is 6450.7kg, which is obtained after 17000 cycles. The design history graph demonstrating the convergence rate of the problem is given in Fig. 5. The results indicate that in the lightest frame the drift constraint for the second floor was 1.024 which is close to its upper bound of 1.17 cm while the lateral displacement of top storey was 5.155 cm against its upper bound of 7.17 cm. The highest ratio among the combined strength constraints was 1.00 which was attained in member 20 which is the outer column of third floor. This clearly indicates that strength constraints are dominant in the optimum design. The frame is also designed using simple hunting search algorithm, particle swarm algorithm, cuckoo search algorithm and big bang- big crunch algorithm developed for unbraced plane frames. Results indicate that second optimum design which is obtained by simple hunting search algorithm is 5% heavier than the one produced by hunting search algorithm with Levy flights modification. These designs are followed by cuckoo search, particle swarm and big bang- big crunch solutions with 6970.60kg, 7532.11kg and 7583.56kg, respectively. These designs are also illustrated in Table 5.

Table 4 Minimum weights (kg) obtained for six storey two bay frame

	HuS-L.F	**HuS**	**CS**	**PS**	**BBBC**
Best run	6450.70	6789.50	6970.60	7532.11	7583.56
Worst run	6979.98	7365.44	7207.79	9240.40	8195.59
Average run	6703.54	6951.53	7098.71	8493.93	7902.15

Table 5 Optimum section designations of six storey two bay frame

Member Group No	Meta-heuristic Search Techniques				
	HuS-LF.	**HuS**	**CS [Saka and Dogan]**	**PS [Saka and Dogan]**	**BBBC [Saka and Dogan]**
1	W610X82	W460X97	W460X82	W530X74	W460X106
2	W460X52	W410X53	W410X53	W310X52	W410X60
3	W200X35.9	W200X35.9	W310X38.7	W200X41.7	W250X49.1
4	W410X67	W460X82	W610X82	W460X89	W360X64
5	W410X53	W360X51	W530X66	W460X89	W360X64
6	W360X44	W360X44	W150X29.8	W360X72	W200X41.7
7	W460X52	W460X52	W460X60	W460X60	W460X60
8	W460X52	W460X52	W460X52	W460X68	W460X60
Max. Int. St. Dr. Ratio	0.88	0.92	0.77	0. 78	0.78
Max. Strength Ratio	1.00	0.98	0.94	0.99	0.98
Top storey drift (cm)	5.155	5.213	4.421	4.5325	4.654
Min. Weight. kg (kN)	6450.70 (63.26)	6789.50 (66.58)	6970.60 (68.358)	7532.11 (73.865)	7583.56 (74.369)

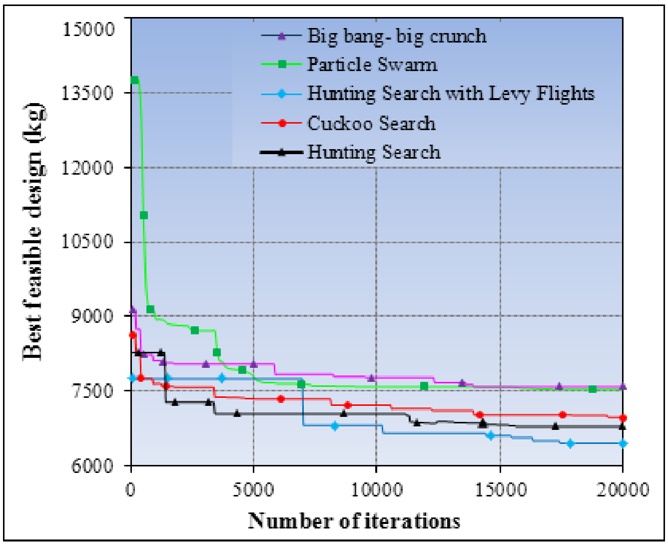

Fig. 5 Design histories of algorithms for six storey two bay frame

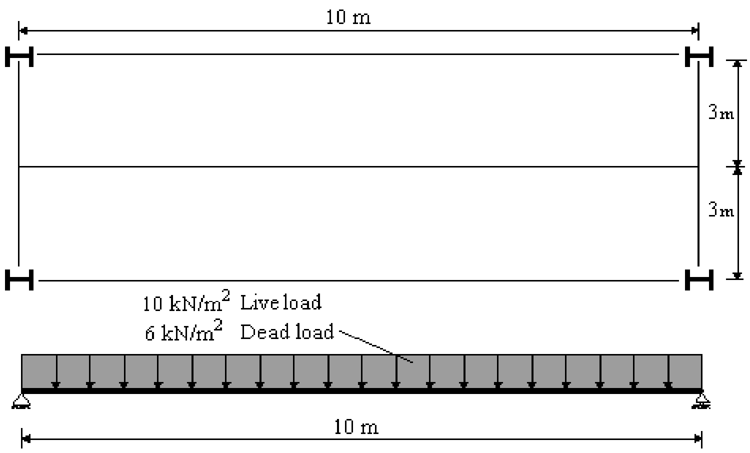

Fig. 6 10-m simply supported beam

5.3 10-m spanned cellular beam

The cellular beam with a span of 10m is designed such that it can carry a uniform dead load of 6kN/m^2 as well as uniformly distributed live load of 10kN/m^2 as shown in Fig. 6. The maximum displacement of the beam under these loads is restricted to 27mm and the modulus of elasticity is taken as 205kN/mm^2. Grade 50 steel is adopted for the steel which has the design strength 355 MPa. Among the steel sections list 64 UB sections starting from 254×102×28 UB to 914×419×388 UB are selected to compose the discrete set of steel profiles from which the design algorithm selects the sectional designations for the cellular beams (Steelwork design guide to BS 5950, 1990). For the hole diameter, which is the second variable of the optimum design problem, discrete set is prepared that has 421 values which starts from 180 mm and goes up to 600 mm with

Table 6 Minimum weights (kg) obtained for 10-m cellular beam

	HuS-L.F	HuS	FF	ABC
Best run	218.1	220.1	221.2	221.2
Worst run	220.8	220.1	227.3	227.3
Average run	218.8	220.1	224.2	225.5

Table 7 Optimum solutions for 10-m cellular beam

Variables	Meta-heuristic Search Techniques					
	HuS-LF	**HuS**	**FF**	**ABC**	**HS** ***Erdal (2011)***	**PS** ***Erdal (2011)***
Opt.UB-Section Designations	UB 305X102X25	UB 305X102X25	UB 305X102X25	UB 305X102X25	UB 305X102X25	UB 305X102X25
Hole Diameter (mm)	386	379	375	375	361	358
Total Number of Holes	21	21	21	21	22	22
Max. strength ratio	0.99	0.98	0.97	0.97	0.96	0.95
Min. Weight (kg)	218.1	220.1	221.2	221.2	221.4	222.1

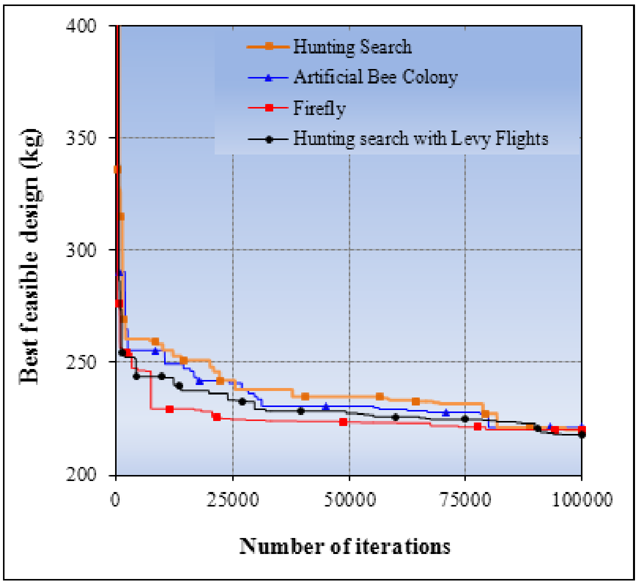

Fig. 7 Design histories of algorithms for 10-m cellular beam

the increment of 1mm. The third discrete set is arranged for the number of holes that contains numbers starting from 2 to 40 with the increment of 1.

Best, worst and average runs obtained by each method are given in Table 6. Results of these four algorithms are presented in Table 7. Hunting search design algorithm with Levy flights selects 305X102X25 UB section for the root beam. This optimum design can be obtained provided that the beam has 26 circular holes each having 386mm diameter. This design with the weight of 218.1kg is recorded as the optimum solution for this example. It is followed by the one produced with simple hunting search with the weight of 220.1kg. Producing a design with the weight of 221.2kg, artificial bee colony algorithm shows the same performance with firefly algorithm. Erdal *et al.* (2011) solved the same example with particle swarm and harmony search (HS) algorithms. These results are also tabulated in Table 7. Design history curves representing the convergence of each method are given in Fig. 7.

6. Conclusions

In this study, it is shown that hunting search algorithm is an efficient and robust technique that can successfully be used in the solution of optimum design problems. Design algorithm is extended to cover the optimum solution of welded beam design problem, cellular beam design problem and unbraced steel plane frame. Further, in order to increase the chance of hunting, search procedure of the algorithm is modified by use of Levy flights. Dimensions of welded beam are treated as design variables of the first example. Sequence number of Universal Beam section, hole diameter and total number of holes in the beam are treated as design variables of cellular beam and sequence number of ready W-section list is treated as design variable of unbraced frame. In addition to hunting search, artificial bee colony, particle swarm, cuckoo search, big bang big crunch algorithms are used for the solution of each design problem. Program results are also compared with the ones taken from literature. Results reveal that hunting search algorithm finds better optimum solutions compare to the other optimum design methods. In addition, it is observed that Levy flights increases the performance of the algorithm. This increase is obtained especially in second example where 5% lighter frame is produced. In the first example, hunting search solutions are followed by particle swarm solution. Cuckoo search produces the third optimum steel frame in second example. In last example, third optimum cellular beam design belongs to firefly algorithm. One common drawback of meta-heuristic search methods is that the iteration process generally gets stuck in the local optima. In this study, it is observed that solutions with constant error strategy experience this fact. However, as can be seen from convergence rate graphs of each example, adaptive error strategy for constraint handling prevents this problem and provides an efficient search for each algorithm.

References

Adami, C. (1998), *An introduction to Artificial Life*, Springer Verlag Telos, London, UK.

Arora, J.S. (2002), *Methods for Discrete Variable Structural Optimization*, *Recent Advances in Optimum Structural Design*, Ed. Burns, S.A., ASCE, USA.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999), *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, U.K.

British Standards BS 5950 (2000), *Structural Use of Steelworks in Building. Part 1. Code of Practice for Design in Simple and Continuous construction, hot rolled sections*, British Standard Institution, London, U.K.

Camp, C.V., Bichon, J.B. and Stovall, S.P. (2004), "Design of steel frames using ant colony optimization", *J. Struct. Eng.*, ASCE, **131**(3), 369-79.

Camp, C.V. (2007), "Design of Space Trusses Using Big Bang-Big Crunch Optimization", *J. Struct. Eng.*, ASCE, **133**(7), 999-1008.

Coello, C.A.C. (2002), "Theoretical and numerical constraint-handling techniques used with evolutionary

algorithms: a survey of the state of the art", *Comput. Meth. Appl. Mech. Eng.*, **191**, 1245-1287.
Dionisio, M.C., Hoffman, R.M., Yost, J.R., Dinehart, D.W. and Gross, S.P. (2004), "Determination of critical location for service load bending stresses in non-composite cellular beams", *17th ASCE Engineering Mechanics Conference*, University of Delaware, Newark, DE.
Deb, K. (1991), "Optimal design of a welded beam via genetic algorithms", *Am. Ins. Aeronaut. Astronaut. J.*, **29**, 2013-2015.
Deb, K. (2000), "An efficient constraint handling method for genetic algorithms", *Comput. Meth. Appl. Mech. Eng.*, **186**(2-4), 311-338.
Doğan, E. and Saka, M.P. (2012), "Optimum design of unbraced steel frames to LRFD-AISC using particle swarm optimization", *Adv. Eng. Softw.*, **46**(1), 27-34.
Duran, O., Perez, L. and Batoccia, A (2012). "Optimization of modular structures using particle swarm optimization", *Exp. Syst. Appl.*, **39**(3), 3507-3515.
Erdal, F., Doğan, E. and Saka, M.P. (2011), "Optimum design of cellular beams using harmony search and particle swarm optimizers", *J. Construct. Steel Res.*, **67**(2), 237-247.
Erdal, F., Doğan, E. and Saka, M.P. (2013), "An improved particle swarm optimizer for steel grillage systems", *Struct. Eng. Mech.*, **47**(4), 513-530.
Erol, O. and Eksin, I. (2006), "A new optimization method: big bang-big crunch", *Adv. Eng. Softw.*, **37**, 106-111.
Fioriti, M. (2014), "Adaptable conceptual aircraft design model", *Adv. Aircraf. Spacecraft Sci.*, , **1**(1), 43-67.
Hasançebi, O. (2008), "Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures", *Comput. Struct.*, **86**, 119-132.
He, S., Prempain, E. and Wu, Q.H. (2004), "An improved particle swarm optimizer for mechanical design optimization problems", *Eng. Optim.*, **36**(5), 585-605.
Karaboga, D. (2010), "Artificial bee colony algorithm", *Scholarpedia*, **5**(3), 6915.
Keane, A.J. (1995), "A brief comparison of some evolutionary optimization methods", *Proceedings of the conference on applied decision technologies (modern heuristic search methods)*, Uxbrigde: Wiley.
Kennedy, J. and Eberhart, R.C. (1995), "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks*, NJ, Piscataway.
Knowles, P.R. (1985), *Design of Castellated beams*, The Steel Construction Institute.
Kochenberger, G.A. (2003), *Handbook of meta-heuristics*, Kluwer Academic Publishers, Norwell, USA.
Lawson, R.M., Lim, J., Hicks, S.J. and Simms, W.I. (2006), "Design of composite asymmetric cellular beams and beams with large openings", *J. Construct. Steel Res.*, **62**, 614-629.
LRFD-AISC (1999), *Manual of steel construction, Load and resistance factor design. Metric conversion of the second edition*, Vol. I & II. AISC, USA.
Mantegna, R.N. (1994), "Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes", *Phys. Rev. E*, **49**(5), 4677-4683.
McGuire, W. (1968), *Steel structures*, Prentice-Hall, Englewood, USA.
Oftadeh, R., Mahjoob, M.J. and Shariatpanahi, M. (2010), "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search", *Comput. Math. Appl.*, **60**, 2087-2098.
Ragsdell, K.M. and Phillips, D.T. (1976), "Optimal design of a class of welded structures using geometric programming", *Am. Soc. Mech. Eng. J. Eng. Indus.*, **98,** 1021-1025.
Saka, M.P. and Doğan, E. (2012), "Design optimization of moment resisting steel frames using a cuckoo search algorithm", *Proceedings of The Eleventh International Conference on Computational Structures Technology*, Dubrovnik, Croatia, September.
Yang, X.S. (2009), *Firefly Algorithms for Multimodal Optimization. Stochastic Algorithms: Foundations and Applications*, SAGA, Lecture Notes in Computer Science, Cambridge, UK.
Yang, X.S. and Deb, S. (2010), "Engineering optimization by cuckoo search", *Int. J. Math. Model. Numer. Opt.*, **1**(4), 330-343.
Ward, J.K. (1990), *Design of composite and non-composite cellular beams*, The Steel Construction Institute Publication, Steel Construction Institute, Ascot, UK.