Steel and Composite Structures, *Vol. 24, No. 1 (2017) 93-112*
DOI: https://doi.org/10.12989/scs.2017.24.1.093

Effect of Levy Flight on the discrete optimum design of steel skeletal structures using metaheuristics

Ibrahim Aydogdu [*1], Serdar Carbas [2a] and Alper Akin [3b]

[1] *Department of Civil Engineering, Akdeniz University, Dumlupinar Bvld., 07058, Antalya, Turkey*
[2] *Department of Civil Engineering, Karamanoglu Mehmetbey University, Karaman, Turkey*
[3] *Trinity Meyer Utility Structures, Memphis, TN, USA*

(*Received October 26, 2016, Revised March 04, 2017, Accepted March 23, 2017*)

Abstract. Metaheuristic algorithms in general make use of uniform random numbers in their search for optimum designs. Levy Flight (LF) is a random walk consisting of a series of consecutive random steps. The use of LF instead of uniform random numbers improves the performance of metaheuristic algorithms. In this study, three discrete optimum design algorithms are developed for steel skeletal structures each of which is based on one of the recent metaheuristic algorithms. These are biogeography-based optimization (BBO), brain storm optimization (BSO), and artificial bee colony optimization (ABC) algorithms. The optimum design problem of steel skeletal structures is formulated considering LRFD-AISC code provisions and W-sections for frames members and pipe sections for truss members are selected from available section lists. The minimum weight of steel structures is taken as the objective function. The number of steel skeletal structures is designed by using the algorithms developed and effect of LF is investigated. It is noticed that use of LF results in up to 14% lighter optimum structures.

Keywords: metaheuristic algorithms; levy flight; steel skeleton structures; artificial bee colony; biogeography-based optimization; brain storm optimization

1. Introduction

Optimum design of steel skeletal structures has always most desired aim of structural engineers. However, until the emergence of numerical optimization techniques this aim has not been fulfilled. The structural design was carried out using trial and error techniques which were mainly based on designers' intuition and experience. The early mathematical optimization techniques were not very capable of finding the solution of discrete optimum design problems. It is after the emergence of stochastic search techniques (metaheuristics) that it became possible to develop discrete optimum design algorithms where the steel design code requirements can be considered in the design and the steel profiles can be selected from available steel sections list. The stochastic search optimization methods are inspired by natural phenomena such as swarm intelligence, survival of fittest, music improvisation, and so forth. Thereby, these techniques owe their success and popularity to being simple, flexible, efficient, and adaptable as well as being easy to apply to complex problems such as real-sized steel structures containing high nonlinearity within itself (Yang *et al*. 2016). Various popular stochastic search algorithms have been introduced recently. For instance, ant colony optimization (ACO) is based on the metaphor of ants seeking food (Dorigo 1992). Particle swarm optimization (PSO) simulates the foraging behavior of a biological social system like a flock of birds (Eberhart and Kennedy 1995). Trying to find a pleasing harmony in a musical performance is analogous to finding the optimum solution in an optimization problem with the harmony search optimization (HSO) (Saka *et al*. 2010). These algorithms have been applied to many engineering optimization problems and proved effective in solving some specific kind of problems such as steel structures (Fourie and Groenwold 2002, Lee and Geem 2004, Perez and Behdinan 2007, Aydogdu and Saka 2009, Carbas *et al*. 2009, Hasançebi and Çarbaş 2011).

Three recent stochastic search algorithms are Biogeography-Based Optimization (BBO) algorithm, Brain Storm Optimization (BSO) algorithm, and Artificial Bee Colony (ABC) algorithm. The BBO algorithm was firstly introduced by Simon in 2008 (Simon 2008), who adopted the theory of island biogeography. The migration and extinction of species between islands is reflected in the mathematical formulation of the BBO. As BBO evolved, it has been implemented to many design optimizations of engineering problems (Roy *et al*. 2011, Jalili *et al*. 2014, Saka *et al*. 2015, Wang *et al*. 2015, Çarbaş 2016). Moreover, different variations of the BBO have been developed to enhance the efficiency of the basic algorithm (Bhattacharya and Chattopadhyay 2010, Gong *et al*. 2010, Yang *et al*. 2013, Aydogdu 2017). Another contemporary trend in swarm based optimization techniques, brain storm optimization (BSO), inspired by brainstorming process in

∗Corresponding author, Ph.D.,
E-mail: aydogdu@akdeniz.edu.tr
[a] Ph.D.
[b] Ph.D.

http://www.techno-press.org/?journal=scs&subpage=6
ISSN: 1229-9367 (Print), 1598-6233 (Online)

human kind, is currently progressing rapidly (Jordehi 2015a). This technique was debut by Shi in 2011 (Shi 2011) and imitates the brainstorming process bringing together a group of people with different backgrounds in order to interactively collaborate for generating eminent thoughts to solve a problem. Despite being one of the latter stochastic optimization techniques, it has a considerable amount of application in different disciplines of optimization problems (Zhan *et al*. 2013, Lenin *et al*. 2014, Li and Duan 2015, Cheng *et al*. 2016). However, its application to the optimum engineering design field as well as in the optimum design of steel structures has not been encountered in the current literature yet. From this aspect, this study will be available in the literature as the first. The last but the most known technique of the triplet is the Artificial Bee Colony (ABC) algorithm, which simulates the intelligent foraging behavior of honeybees (Karaboga 2005). The ABC as an optimization tool provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees with time and the bees' purpose is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. This technique has been proved to be very robust and effective in finding the solutions of different types of discrete programming problems such as steel skeleton structures (Hadidi *et al*. 2010, Sonmez 2011, Degertekin 2012).

A Levy Flight (LF) is a class of random walk generalized Brownian motion to include non-Gaussian randomly distributed step sizes for the distance moved (Al-Temeemy *et al*. 2010). There are many natural and artificial facts that may be depicted by LF, such as fluid dynamics, earthquake analysis, the diffusion of fluorescent molecules, cooling behavior, noise, etc. LF is also used in the field of ultrasound in skin tissue (Pereyra and Batatia 2010) and in radar scanning (Chen 2010). LF also plays a significant role in many fields such as computer sciences (Terdik and Gyires 2009). In the current paper the LF is combined with the BBO, BSO, and ABC algorithms so that their performance on searching through global optima is intensified. By preventing divergence, each technique is also capable of solving highly nonlinear problems such as steel frames and trusses. The algorithms integrated with LF select the sequence number of W-shape and pipe shape steel sections listed in steel profile table which are managed as design variables. The displacement limitations, inter-story and top-story drift restrictions, ultimate strength ratios, and the geometric necessities are treated as design constraints which are enforced according to the specifications of LRFD-AISC (Load and Resistance factor Design – American Institute of Steel Construction) (LRFD 2000). The effectiveness of LF integrated algorithms are compared to those of standard versions in order to reveal the outstanding performance of the proposed strategy in design optimization of two real-sized steel frames and a steel barrel vault.

The remainder of this manuscript is organized as follows; In Section 2, the mathematical formulations of discrete design optimization is depicted for steel frames and trusses according to LRFD-AISC under pre-described constraints. In Section 3, the basic steps of the BBO, BSO, and ABC techniques are outlined. In Section 4, the LF strategy is identified in detail. In Section 5, computational procedure of the optimization algorithms are given. Performance of the Levy Flight-based metaheuristics on mathematical benchmark functions is tested in Section 6. In Section 7, the efficiency and accuracy of the proposed LF integrated algorithms are investigated in solving selected real-sized design examples, namely two steel frames and a barrel vault, by comparing the results of those derived from basic algorithms. Also a set of results obtained from optimum designs for numerical solutions of design examples are presented and discussed in this section. Sensitivity analysis of the control parameters are illustrated in Section 8. Finally, concluding remarks are provided in Section 9.

2. Discrete optimization of steel skeletal structures

2.1 Mathematical modelling of the optimization problem

The discrete optimization procedure of the steel skeletal structures can be defined as searching optimum steel sections for grouped structural members in order to minimize the weight of the structure. For steel structures, the objective function can be taken as the minimum weight of the structure to observe the overall economy or the material cost of the structure while behavioral and geometrical constraints are satisfied according to the design specification.

Hence, the discrete optimum design problem of steel skeletal structures can be formulated as:

Find the steel sections of the optimum design

$$\boldsymbol{x} = [x_1, x_2, \dots x_{NG}]^T \tag{1}$$

In order to minimize the weight of the structure

$$W(\boldsymbol{x}) = \sum_{i=1}^{NM} w_i \cdot l_i \tag{2}$$

where, x is the vector of integer values representing the sequence numbers of steel sections assigned to member groups, NG *and* NM respectively the total member groups and number of members defined in the structure, $W(x)$ is the total weight of the structure, w_i is the unit weight of the selected steel section to be adopted for the structural member i, and l_i is the length of member i.

Subjected to the following constraints:

- Strength constraints for the beam-column members of the structures (LRFD 2000)

$$g_1(\boldsymbol{x}) = \left(\frac{P_u}{\emptyset_c P_n} + \frac{8}{9}\left(\frac{M_{ux}}{\emptyset_b M_{nx}} + \frac{M_{uy}}{\emptyset_b M_{ny}}\right)\right)_{i,j} - 1.0 \le 0 \quad \text{for} \quad \left(\frac{P_u}{\emptyset_c P_n}\right)_{i,j} \ge 0.2 \tag{3}$$

$$g_1(\boldsymbol{x}) = \left(\frac{P_u}{2\emptyset_c P_n} + \left(\frac{M_{ux}}{\emptyset_b M_{nx}} + \frac{M_{uy}}{\emptyset_b M_{ny}}\right)\right)_{i,j} - 1.0 \le 0$$

$$\text{for} \quad \left(\frac{P_u}{\emptyset_c P_n}\right)_{i,j} < 0.2$$

$$i = 1, 2, \dots NM; \quad j = 1, 2, \dots N_{lc}$$

In Eq. (3), P_u represents the ultimate axial load occurred in the member i under the load case j, P_n represents the axial load capacity of the member i, M_{ux} *and* M_{uy} are the ultimate moments member i is subjected to under the load case j for the local x and y axis respectively, N_{lc} is the total number of load cases, and ϕ_c *and* ϕ_b are the safety factors of the axial load and bending capacities, respectively. P_n and P_u can be the tension or compression. For determinations of the M_{ux} and M_{uy} values, second order (P-Δ) effects which are alternatively calculated according to section C1 of the LRFD-AISC (LRFD 2000). In the alternative calculation, two ultimate moments are obtained from the superposition of the results of non-sway and sway analyses of the structure. The first moment (M_{nt}) is calculated from the non-sway analysis of the structure under the gravity loads. The second moment (M_{lt}) is calculated from the sway analysis of the structure under the lateral loads. These moments are combined using magnifier coefficients and the ultimate moment is calculated as follows

$$M_u = B_1 M_{nt} + B_2 M_{lt} \tag{4}$$

where, B_1 and B_2 respectively are the magnifier coefficients of M_{nt} and M_{lt}. The details of how these coefficients are calculated are given in Chapter C of LRFD-AISC. For the truss structure (3rd example), P-Δ effect is not taken into account.

- Deflection constraints for all members of the frame structures

$$g_2(\boldsymbol{x}) = \frac{(\delta)_{i,j}}{(\delta_{al})_i} - 1.0 \le 0 \tag{5}$$

$$i = 1,2,\dots,NM; \quad j = 1,2,\dots,n_{lc}$$

where, $(\delta)_{i,j}$ is the deflection of the i^{th} frame member under the load case j and $(\delta_{al})_i$ is the allowable deflection limit for member i.

- Top-story and inter story drift constraints for the frame structures

$$g_3(\boldsymbol{x}) = \frac{(\Delta^{top})_j}{\Delta^{top}_{al}} - 1.0 \le 0; \quad j = 1,2,\dots,n_{lc} \tag{6}$$

$$g_4(\boldsymbol{x}) = \frac{(\Delta^{is})_{i,j}}{\left(\Delta^{is}_{al}\right)_i} - 1.0 \le 0 \tag{7}$$

$$i = 1,2,\dots,n_{st}, \quad j = 1,2,\dots,n_{lc}$$

where, $(\Delta^{top})_j$ is the maximum top story drift under the j^{th} load case, Δ^{top}_{al} is the allowable deflection top story drift limit of the structure, $(\Delta^{is})_{i,j}$ is the maximum inter story drift between upper and lower joints of the i^{th} story under the j^{th} load case, $\left(\Delta^{is}_{al}\right)_i$ is the allowable inter story drift limit of the i^{th} story, and n_{st} is the total number of stories in the structure.

Displacement constraints for all joints of the truss structure

$$g_5(\boldsymbol{x}) = \frac{(\delta i)_{j,l}}{\delta i_{al}} - 1.0 \le 0 \tag{8}$$

$$j = 1,2,\dots,n_j, \quad l = 1,2,\dots,n_{lc}$$

where, $(\delta i)_{j,l}$ is the displacement of the j^{th} joint under the load case l and δi_{al} is the allowable displacement limit. In Eqs. (5)-(8), the allowable deflection, displacement and drift values are computed in accordance with the ASCE Ad Hoc Committee report (Ellingwood 1986).

- Geometric constraints for the column to column (CtoC) and beam to column (BtoC) connections of the frame structures

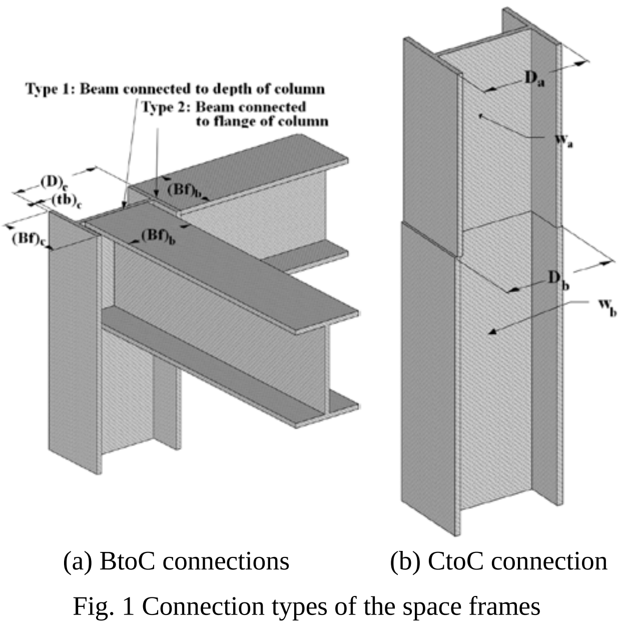
$$g_6(\boldsymbol{x}) = \sum_{i=1}^{n_{cc}} \left(\frac{D_i^a}{D_i^b} - 1.0\right) \le 0 \tag{9}$$

$$g_7(\boldsymbol{x}) = \sum_{i=1}^{n_{cc}} \left(\frac{w_i^a}{w_i^b} - 1.0\right) \le 0 \tag{10}$$

$$g_8(\boldsymbol{x}) = \sum_{i=1}^{n_{bc1}} \left(\frac{Bf_i^b}{D_i^c - 2 \cdot tf_i^c} - 1.0\right) \le 0 \tag{11}$$

$$g_9(\boldsymbol{x}) \sum_{i=1}^{n_{bc2}} \left(\frac{Bf_i^b}{Bf_i^c} - 1.0\right) \le 0 \tag{12}$$

where, D_i^a and w_i^a are respectively the depth and the unit weight of the top column in the i^{th} CtoC connection, D_i^b and w_i^b respectively are the depth and unit weight of the



(a) BtoC connections (b) CtoC connection

Fig. 1 Connection types of the space frames bottom column in the i^{th} CtoC connection, n_{cc} is the total number of CtoC connections in the structure, Bf_i^b is the flange width of the beam, D_i^c and tf_i^c respectively are the depth and flange thickness of the column in the BtoC connection type 1, Bf_i^c is the flange width of the column in the BtoC connection type 2, n_{bc1} and n_{bc1} are respectively the total numbers of the type1 BtoC connection and the type 2 BtoC connection. In order to illustrate geometric constraints clearly, CtoC and BtoC connection types are illustrated in detail in Fig. 1.

2.2 Constraint handling and evaluation of the objective function

Many alternative constraint handling techniques for metaheuristic algorithms are available in the literature: static penalty function (Homaifar *et al*. 1994), dynamic penalty function (Joines and Houck 1994), adaptive penalty function (Ben Hadj-Alouane and Bean 1997), repair approaches (Michalewicz and Nazhiyath 1995), separatist approaches (Surry *et al*. 1995). Review studies about constraint handling techniques are also available in the literature (Coello and Carlos 1999, Coello 2002, Salcedo-Sanz 2009, Jordehi 2015e). Among these approaches, static penalty function, whose efficiency was previously proved in the discrete optimization of skeleton structures (Aydogdu and Saka 2009, Carbas *et al*. 2009, Aydoğdu 2010, Aydoğdu *et al*. 2012a, b, Aydoğdu and Akın 2014, Artar and Daloglu 2015b, Artar and Daloglu 2015a, c, Yetkin 2015, Carbas and Aydogdu 2017), is used in this study. Application of the static penalty function method for the discrete optimization of steel skeleton structures can be described as follows:

Each solution where the set of steel sections are assigned as design variables, the structural analysis is performed and responses of each candidate solution are obtained under the applied loads.

The total violation, *V*, is calculated using Eq. (13) for infeasible solutions that violate some of the problem constraints described in Eqs. (3)-(12).

$$V = \sum_{i=1}^{NC} C_i \,; C_i = \begin{Bmatrix} 0 \;\; \text{for} \;\; g_i(\boldsymbol{x}) \le 0 \\ g_i(\boldsymbol{x}) \;\; \text{for} \;\; g_i(\boldsymbol{x}) > 0 \end{Bmatrix} \tag{13}$$

In Eq. (13), C_i is the violation of the i^{th} problem constraint: $g_i(x)$ and *NC* represents the total number of constraints.

If the total violation (*V*) is greater than zero, structure weight is penalized using penalty function described in Eq. (14).

$$W_p = W \cdot (1 + V)^2 \tag{14}$$

where, W_p is penalized weight of the structure. If *V* is calculated as zero that means structure satisfies all problem constraints, penalized weight of the structure directly equals to its unpenalized (real) weight.

3. ABC, BBO and BSO algorithms

3.1 ABC algorithm

The ABC optimization algorithm is developed by Karaboga and Basturk (Karaboga 2005, Karaboga and Basturk 2007, 2008) which is adopted from the behavior of bee swarms. Three types of worker bees are included in the bee swarms which perform different tasks. These are employed bees, onlooker bees and scout bees. The employed bees determine the location and nectar capacity of flowers. After return to the hive, these bees share this information with the onlooker bees by performing the waggle dance. The onlooker bees watch the dance and fly to the food source which has a rich amount of nectar. If the food source is exhausted, the scout bees randomly search for new food sources. Bees go to one food source during each trip. Therefore, the number of employed bees and onlooker bees are equal to the number of the food sources. If the food source is exhausted, scout bees replace the onlooker and the employed bees which go to the exhausted food source.

For structural optimization, ABC algorithm is utilized to find the optimum steel sections assigned to member groups of the structure with the purpose of minimizing the weight of the structure. The location of the food sources (*FS*) represents the structural designs, each coordinate of the location represents the design variable of the structure and the nectar amount of the food source represents fitness values of the structures. Main steps of the ABC method for an optimization problem are described as follows:

Step 1: In this step, initial structural designs (*FS* designs) are generated randomly using Eq. (15). Then, the designs are evaluated and their fitness values are computed using Eq. (16).

$$X_{ij} = \mathrm{int}(1 + (\mathrm{NSec} - 1) \cdot \mathrm{rnd}) \qquad i = 1, \ldots, FS; \quad j = 1, \ldots, NG \tag{15}$$

$$F_i = \frac{1}{(W_p)_i}; \qquad i = 1, \ldots, FS \tag{16}$$

In these equations, *NSec* represents the total number of steel sections which are adopted from LRFD-AISC, $\boldsymbol{X}$ matrix of integer values represents the sequence numbers of steel sections for all designs in the algorithm memory and *rnd* is a random number between 0-1. The evaluated structural designs, their penalized weights and their fitness are stored in the algorithm memory.

Step 2: The employed bees generate new structural designs (x^{Cand}) by modifying the previous designs in the memory which is described as follows

$$x^{Cand}{}_j = X_{ij} + 2 \cdot (rnd - 0.5) \cdot (X_{ij} - X_{kj}) \qquad i, k = 1,2, \ldots, FS, \quad j = 1,2, \ldots, NG \tag{17}$$

where *i* and *k* respectively represent indexes of the previous food source (structural design) and neighbor of i^{th} food source. The new designs are evaluated and their fitness values are calculated by using Eq. (16). If the fitness value of the new design is better than the fitness value of the former design, the former design is replaced by the new design. This process is named as "Greedy Selection". If the replacement is not performed, the trial number of the food source is increased by one. After the greedy selection procedure, selection probabilities of the designs in the memory are calculated as follows

$$P_i = \frac{F_i}{\sum_{i=1}^{FS} F_i}; \qquad i = 1,2,\ldots,FS \tag{18}$$

Step 3: The onlooker bees decide which former design is used for the generation of the new design. The decision criterion of this process is described as follows

$$\text{if} \quad P_i > rnd \qquad \textit{The design is selected} \tag{19}$$

Then, the onlooker bees generate the new structural design and apply the procedures in the same fashion as the employed bees.

Step 4: The trial numbers of the food sources are checked in this step. If the trial number exceeds the limits of the food source (*LFS*), the structural designs of the food source is discarded and scout bees randomly generate the structural designs (finds new food sources) in place of former designs by using Eq. (15).

Steps 2 to 4 are repeated until a pre-assigned maximum number of iterations are completed.

3.2 BBO algorithm

The BBO algorithm was initially developed by Simon (Simon 2008) and is based on the geographical behavior of individuals in the habitat such as migration, existence and extinctions. In the algorithm, two main parameters, *HSI* (high suitability index) and *SIV* (suitability index variable), control these behaviors. *HSI* is related to the life conditions of the islands which can be modeled as fitness value of the solution vector. *SIV* describes habitability of individuals in the islands, which is independent design variable of the solution vector.

The mathematical modeling of the algorithm consists of two main phases; migration and mutation. In the migration phase, individuals move from one habitat to another, which means generation of new solution vectors by modifying former solutions. The movements are performed by using the roulette wheel selection method. The movement probabilities of individuals are determined using their immigration and emigration rates which are related to the fitness values of the solution vectors. In mutation phase, the mutation probabilities of all the individuals are determined first. If the mutation takes place, any design variable of the individual is randomly changed.

For structural optimization, each habitat represents structural design and the individuals in the habitat represent design groups (variables) of the structure. The BBO algorithm is described in detail as follows:

Step 1: Initial habitats (structural designs) are generated randomly using Eq. (15). The number of initial designs is equal to the number of habitats (*NH*). Then, some procedures are applied in same way described in the step 1 of the ABC algorithm.

Step 2: The migration phase is performed in this step. First, structural designs are sorted ascending order emigration (μ) and immigration (λ) rates of the designs are calculated as follows

$$\mu_i = \frac{NH + 1 - i}{NH + 1}; \quad \lambda_i = 1 - \mu_i; \quad i = 1,2,\ldots,NH \tag{20}$$

Then, the new design is generated by changing the former designs according to μ and λ. The generation process can be described in simple pseudo code as follows

If ($rnd < \lambda_k$) ***Then***
  Do $j = 1, NG$
    $RandN = rnd * \sum_{i=1}^{NH} \mu_i$
    $Select = \mu_1$
    Select Index = 1
    Do While (*RandN* > *Select.and.SelectIndex* <*NH*) (21)
      SelectIndex = *SelectIndex* + 1
      $Select = Select + \mu_{SelectIndex}$
    End Do
$$x^{Cand}_j = X_{SelectIndex,j}$$
  End Do
End if

Step 3: Selected designs are mutated in this phase. Selection criteria of the designs are dependent on their mutation rates. Mutation rates and selection criteria of the designs are calculated as follows

$$m_i = m_{\max}\left(\frac{1 - P_i}{P_{\max}}\right); \quad i = 1,2,\ldots,NH \tag{22}$$

$$\text{If} \quad rnd < m_i, \quad \text{mutation is performed} \tag{23}$$

where, P_i is a selection probability of the i^{th} habitat (design) (Simon 2008). In the mutation process, randomly determined group of the structural design is modified randomly in the same way described in Eq. (15).

At the end of the step 3, elite designs having best solutions are stored to use next generations. Steps 2 to 3 are repeated until a pre-assigned maximum number of iterations are completed.

3.3 BSO algorithm

BSO algorithm is a recent stochastic search algorithm developed via using the simulation brain storm activity of a group of people in order to generate great ideas for the solution of their problem. In the algorithm, people with different knowledge background gather and generate a group. In the group, people generate different ideas resulting in an idea database. The ideas are evaluated according to their greatness and clustered in subgroups. Subsequently, new ideas are generated using the idea database and the knowledge background. These procedures continue until the satisfactory great ideas are found.

In the algorithm, the idea represents structural design and the greatness of the idea represents the fitness value of the structure for the current optimization problem. General steps of the BSO algorithm are described as follows:

Step 1: Generation and evaluation of initial designs (number of individuals (NI)) are performed the same way as step 1 of the ABC and BBO algorithms.

Step 2: Initial designs are clustered into subgroups. K-means method is used for the clustering. Then, the designs in each cluster are sorted in descending order with respect to their fitness value and the best designs in each cluster are defined as the center of the cluster.

Step 3: The centers of clusters is modified randomly according to the pre-determined probability (Ps_3). Then, the selection probabilities of clusters, based on the number of individuals in the clusters, are calculated as follows

$$P_{Cl,i} = \frac{NI_i}{NI}; \quad i = 1,2,\ldots,NCl \tag{24}$$

Step 4: The candidate designs are generated in this step through either using one cluster or two clusters as described below

If ($rnd < Ps_{4,1}$) ***Then***
The design is generated using one cluster.
Else (25)
The design is generated using two cluster
End if

where $Ps_{4,1}$ represents pre-determined probability used for selecting one cluster to generate new design. If one cluster is being used, the cluster in the group is randomly selected. If the selection probability of the cluster (P_{Cl}) is lower than random value (rnd), the cluster is discarded and another cluster is selected randomly. After selection of the cluster, the candidate design is generated as follows

Randomly select one cluster j
If ($rnd < Ps_{4,2}$) ***Then***
$x^{cand}_k = Center_{j,k}$; $k = 1, 2,\ldots, NG$
Else (26)
Select any individual ($i1$) *from the cluster j*
$x^{cand}_k = X_{i1,k}$; $k = 1, 2,\ldots, NG$
End if

where $Ps_{4,2}$ is usage probability for the center of the selected cluster. If two clusters are being used, the clusters in the group are randomly selected. Then, the candidate design is generated as follows

Randomly select two clusters; $j1$ *and* $j2$
If ($rnd < Ps_{4,2}$) ***Then***
$x^{cand}_j = rnd*Center_{j1,k}+(1-rnd)*Center_{j2,k}$
$k = 1, 2,\ldots, NG$
Else (27)
Select any individual ($i1$) *from the cluster* $j1$
Select any individual ($i2$) *from the cluster* $j2$
$x^{cand}_k = rnd*X_{i1,k}+(1-rnd)* X_{i1,k}$; $k = 1, 2,\ldots, NG$
End if

Step 5: Generated candidate designs are modified in this step to increase diversity. In this process, step length (SL) values are added to the candidate designs. Calculation of the SL values and update procedure of the candidate designs are described as follows

$$SL = logsig((0.5Iter_{\max} - Iter)/20)*rnd \tag{28}$$

$$x^{cand}_k = x^{cand}_k + SL*6*(rnd - 3) \tag{29}$$

where SL is the step length value used for generating the new individual, $Iter$ is the current iteration number in the optimization process and $Iter_{\max}$ *is* the maximum iteration number defined in the optimization method. After addition of the SL values, greedy selection procedure is applied as described in step 2 of the ABC algorithm.

Steps 2 to 5 are repeated until a pre-assigned maximum number of iterations are completed.

4. Levy Flight strategy

4.1 Mathematical background

LF, also called Levy motion, demonstrated a type of non-Gaussian stochastic process whose step size is distributed based on a Levy stable distribution (Levy 1939). When generating new solution x^{t+1} for solution i, a LF is performed.

$$x^{t+1} = x^t + \alpha \oplus Levy(\beta) \tag{30}$$

where $\alpha > 0$ is the step size which is relevant to the scales of the problem and β is stability (Levy) index. In most conditions, we let $\alpha = 1$. The product $\oplus$ means entry wise multiplications (Yang and Deb 2009). LF essentially provides a random walk while its random step is drawn from a Levy distribution for large steps as depicted in the following

$$\alpha \oplus levy(\beta) \sim 0.01 \frac{u}{v^{\frac{1}{\beta}}} (x_j - x_j^{best}) \tag{31}$$

where u and v values are obtained from normal distributions

$$u \sim N(0, \sigma_u^2), \qquad v \sim N(0, \sigma_v^2) \tag{32}$$

with

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta\, 2^{\frac{\beta-1}{2}}} \right]^{1/\beta} \quad \sigma_V = 1 \tag{33}$$

where Γ is 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. That is, if k is a positive integer $\Gamma(k)=(k-1)!$

There are a few ways to implement LFs; the method chosen in this paper is one of the most efficient and simple ways based on the Mantegna algorithm; all the equations are detailed in Refs. (Yang 2010, Aydoğdu *et al*. 2016).

4.2 Application of LF strategy on the optimization techniques

The main property of the stochastic search techniques is the usage of randomness for generating the new solution. Most of these techniques generate initial solutions randomly and improve the solutions using characteristic formulas contain randomness. Some of the stochastic search techniques have additional randomness parts in order to prevent local convergence. Although this method performs well in many optimization problems, it can fail in the current optimization problem due to the optimization problem containing discrete design variables and irregular constraint functions (especially geometric constraints). Therefore, in progressive iterations, solutions which are generated in the randomness parts are not necessarily better solutions than previous solutions. This situation causes divergence in the algorithm. Although, adjustment of the search parameters is an alternative way to overcome this problem, this strategy can be inadequate in the current problem. LF strategy is based on modifying old solutions by using random walk strategy. Step size of the random walk can be adjusted according to the scale of the optimization problem. This makes it convenient to balance exploration and exploitation. Therefore, solutions obtained using LF can likely be better than the solutions generated randomly.

In the study, the LF strategy is used instead of the randomness parts of the presented algorithms. For ABC optimization method, the LF strategy is used in step 4 (randomly generation formula of scout bees). For BBO algorithm, the LF strategy is used in step 3 (randomly generation formula of mutation process). For BSO algorithm, the LF strategy is used in step 3 (randomly modification of center of clusters). In order to prevent the local convergence in the improved versions of the optimization algorithms using LF strategy, two solutions have been developed. The first one is adjusting step size which is related to LF index. Aydogdu *et al*. (2016) performed the sensitivity analysis for the optimum design of real size steel frames. According to the study, LF index is determined as 1.5. The second solution is intended to avoid redundant solutions in the algorithm memory.

5. Computational procedure of the optimization algorithms

The applications of the artificial bee colony optimization, bio-geography based optimization and brain storm optimization algorithms to the problem of the steel skeletal structures are respectively summarized in the following subsections.

5.1 Optimum design algorithm of the ABC

The design algorithm of Artificial Bee Colony (ABC) technique consists of the following steps (Carbas *et al*. 2013)

(1) Select the values of the ABC algorithm parameters. These are number of employed bees, number of onlooker bees, number of cycles and control parameter adjusting the food source. In the algorithm, the number of employed bees and onlooker bees are equal to the number of the food sources.

(2) After defining search parameters, all foragers in the colony search food source randomly. This means, the randomly generated number of the steel skeletal structure is equal to the sum of the number of employed bees and the number of onlooker bees. Then, generated structural designs are evaluated and penalized in accordance with their weights and constraints violations.

(3) After evaluation process, bees having the best structural designs become employed bees. Then, employed bees start to generate a new structural design by using the old one.

(4) After finding new structural designs and replacements, all employed bees return their hive and start their waggle dance. Waggle dance of employed bees are related to penalized weight of structural designs. The remainders of the bees (onlooker bees) watch the waggle dance and make a decision. This decision process of each onlooker bee depends on its probability value associated with the structural design.

(5) If steel structural design cannot be replaced with the old design, this structural design is abandoned and the employed bee associated with that design becomes a scout bee. Scout bees generate new structural designs by using random selection process the same as step 2.

(6) Steps 3 and 5 are repeated until the termination criterion is satisfied; that is the pre-selected maximum number of iterations is reached. This number is selected large enough such that within this number of design iterations no further improvement is observed in the weight of the steel skeletal structure.

5.2 Optimum design algorithm of BBO

The design algorithm of bio-geography based optimization consists of the following steps (Simon 2008);

(1) Initialize the BBO parameters. This means deriving a method of determining problem solutions to SIVs and habitats. Also, initialize the maximum species count, the maximum migration rates, the maximum mutation rate, and an elitism parameter.

(2) Initialize a random set of habitats, each habitat corresponding to a steel skeleton structure design to the given problem.
(3) For each habitat, determine the HSI to the number of species, the immigration rate, and the emigration rate.
(4) Probabilistically use immigration and emigration to modify each non-elite habitat.
(5) For each habitat, update the probability of its species count. Then, mutate each non-elite habitat based on its probability, and recalculate each HSI.
(6) Go to step (3) for the next iteration. This loop can be terminated after a predefined number of generations.

5.3 Optimum design algorithm of BSO

The design algorithm of the BSO consists of the following steps (Shi 2011);

(1) Initialize the BSO parameters. These are the number of individuals, the number of clusters, the probability of selecting one cluster for creating the new individual, the probability of randomly replacing a cluster center, and the probability of using the cluster center.
(2) Initialize individuals randomly where each individual represents the steel skeleton structure designs in the problem. Evaluate the steel skeleton structure designs. Sort the steel skeleton structure designs in ascending order of their penalize weight.
(3) Cluster individuals into m clusters; determine center and selection probabilities of m clusters.
(4) If the randomly generated number between 0 and 1 is smaller than a pre-determined probability of randomly replacing a cluster center then,
    (a) Randomly select a cluster;
    (b) Randomly generate an individual to change the selected cluster center;
(5) If randomly generated number between 0 and 1 is less than the probability of selecting one cluster then
    (a) Pick one cluster according its selection probability
    (b) If randomly generated number between 0 and 1 is less than the probability of using the cluster center then,
        (i) Generate the steel skeleton structure design according to the selected cluster center
    Else
        (ii) Generate the steel skeleton structure design according to any individual in the selected cluster
    Else
    (c) Pick two clusters randomly
    (d) If randomly generated number between 0 and 1 is less than the probability of using the cluster center then,
        (i) Generate the steel skeleton structure design according to the centers of the selected clusters
    Else
        (ii) Generate the steel skeleton structure design according to any two individuals in the selected clusters
(6) Modify the steel skeleton structure design using the step length
(7) Evaluate the steel skeleton structure design and compare to the existing individual with the same individual index; the better one is kept and registered as the new individual;
(8) If n new individuals have been generated, go to step 9; otherwise, go to step 5;
(9) Terminate if the pre-determined maximum number of iterations has been reached; otherwise go to step 3.

6. Performance of the Levy Flight-based metaheuristics on mathematical benchmark functions

Prior to applying the developed algorithms (LFABC, LFBBO and LFBSO) on the discrete optimization of the steel skeletal structures, these algorithms are tested on four mathematical benchmark functions. Specifications of the benchmark functions are given in the Table 1.

In the table, n represents the dimension of the function which is equal to number of decision variables in the optimization problem. In this study, three different n values have been used (n = 5, n = 30, n = 100). Obtained results of the examples are compared to the standard versions of the presented algorithms (ABC, BBO and BSO) as well as the following well-known optimization algorithms; Conventional Particle Swarm Optimization (CPSO) (Shi and Eberhart 1998), Harmony Search Optimization (HSO) (Geem *et al*. 2001), Firefly Algorithm (FFA), Genetic Algorithm (GA) (Digalakis and Margaritis 2002), Gravitational Search Algorithm (GSA) (Rashedi *et al*. 2009), and Enhanced Leader Particle Swarm Optimization (ELPSO) (Jordehi 2015d). In the optimization procedure, the number of function iterations is kept the same as the compared algorithms. In LFABC algorithm, the number of food sources are respectively taken as 5**n* and 10**n*. In LFBBO and BBO algorithms, the number of habitat and the maximum mutation rate is respectively set as 5**n* and

Table 1 Specifications of the benchmark functions

Func. name	Formulation	Range
Sphere	$F_1(\boldsymbol{x}) = \sum_{i=1}^{n}\left(x_i^2\right)$	[-5.12, 5.12]
Rastrigin	$F_2(\boldsymbol{x}) = 10n + \sum_{i=1}^{n}\left(x_i^2\text{-}10\cos(2\pi x_i)\right)$	[-5.12, 5.12]
Griewank	$F_3(\boldsymbol{x}) = \sum_{i=1}^{n}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{n}\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]
Rosenbrock	$F_4(\boldsymbol{x}) = \sum_{i=1}^{n\text{-}1}\left(100(x_i^2 + x_{i+1})^2 + (x_i - 1)^2\right)$	[-5, 10]

Table 2 Statistical data of the algorithms on mathematical benchmark functions (n = 5) (Best values are bolded)

F_1	LFABC	LFBBO	LFBSO	ABC	BBO	BSO	ELPSO	CPSO	HSO	GA	FSO	GSA
Mean	6.87e-10	1.05e-09	3.75e-16	5.28e−11	6.99e-05	2.29e−4	**3.20e−15**	2.176e−13	5.86e−5	3.95e−5	3.73e−7	4.53e−7
Std	8.62e-10	2.58e-09	1.43e-16	5.08e−11	6.54e-05	1.27e−4	**3.10e−15**	1.936e−13	1.031e−4	3.57e−5	3.09e−5	1.432e−6
Median	4.86e-10	2.40e-10	3.74e-16	3.41e−11	4.35e-05	2.24e−4	**2.02e−15**	1.55e−13	1.77e−5	2.90e−5	3.17e−5	0
Min	2.42e-11	3.60e-12	1.58e-16	4.2e−12	5.31e-06	3.42e−5	**1.5e−16**	3.43e−14	1e−7	2.1e−6	6e−6	0
Max	4.40e-09	1.37e-08	6.53e-16	1.780e−10	0.000307	4.84e−4	**1.15e−14**	9.984e−13	5.188e−4	1.648e−4	1.546e−6	4.528e−6
F_2												
Mean	**3.19e-06**	1.21	1.21	8.43e−6	0.0131	0.975	0.0141	1.9237	0.0215	1.6921	1.791	0.3449
Std	**3.15e-06**	1.16	0.943	8.43e−6	0.0136	0.637	0.0083	1.043	0.0373	1.1018	0.9143	0.5652
Median	2.04e-06	0.995	0.995	4.84e−6	0.0101	1.04	0.0199	1.9899	0.0051	1.5729	1.99	**0**
Min	3.89e-07	1.44e-09	**0**	1.2e−7	0.000554	0.0231	**0**	**0**	**0**	0.1346	0.0001	**0**
Max	**1.49e-05**	4.97	2.98	3.169e−5	0.0651	2.17	0.0798	3.9798	0.1372	3.9168	2.985	1.4171
F_3												
Mean	0.0261	0.161	6.40	0.0136	0.0694	0.687	**0.0043**	0.0927	0.3492	0.1189	0.03	7.4448
Std	0.0132	0.103	5.06	0.0065	0.0256	0.418	**0.0014**	0.0589	0.2301	0.0405	0.0218	3.3767
Median	0.0243	0.152	4.88	0.0137	0.0626	0.643	**0.0029**	0.0714	0.2938	0.1152	0.0229	6.7265
Min	0.00875	0.01232	0.952	0.0003	0.0272	0.163	**0.0007**	0.0105	0.0023	0.0441	0.0101	3.1479
Max	0.0618	0.434	24.2	0.0292	0.139	1.59	**0.034**	0.2587	0.7713	0.1921	0.0948	13.06
F_4												
Mean	0.393	2.13	0.791	0.0746	1.32	0.618	**0.0551**	0.2621	3.3874	1.1055	2.0458	5.7634
Std	0.495	0.796	0.737	0.0829	0.913	0.318	**0.0239**	0.9973	2.5074	0.6436	1.3827	1.7891
Median	0.185	2.20	0.591	0.044	1.36	0.6733	**0**	**0**	3.565	1.1506	1.8992	5.4878
Min	0.00458	0.604	0.000490	0.0081	0.00482	0.0928	**0**	**0**	0.0357	0.0373	0.1552	2.9368
Max	1.848	4.47	3.51	0.3477	2.54	1.24	**0.1308**	3.9308	11.5375	2.6914	4.5985	9.4227

Table 3 Statistical data of the algorithms on mathematical benchmark functions (n = 30) (Best values are bolded)

F_1	LFABC	LFBBO	LFBSO	ABC	BBO	BSO	ELPSO	CPSO	HSO	GA	FSO	GSA
Mean	0.0579238	107.4762	**4.66e-11**	0.4114	0.0006317	0.0419	5.244e−8	6.11e−8	1.0389	2.0586	1.819e−5	0.0237
Std	0.207686	20.88617	**1.15e-11**	0.281	0.0002194	0.0051	1.643e−8	3.64e−8	0.5401	0.5491	4.92e−6	0.0019
Median	0.0042	106.7063	**4.637e-11**	0.3935	0.0006021	0.0416	5.176e−8	5.92e−8	0.8989	2.086	1.721e−5	0.023
Min	0.0001575	75.95851	**2.003e-11**	0.0265	0.0003066	0.0321	2.979e−8	1.36e−8	0.555	1.0484	9.79e−6	0.0221
Max	1.1222	143.6848	**7.687e-11**	1.2233	0.0011622	0.05	7.614e−8	1.500e−7	1.9489	2.9578	3.303e−5	0.0259
F_2												
Mean	10.0972	301.8013	28.5553	62.841	**0.1171841**	42.7192	8.6403	25.67	24.3984	142.0224	29.7863	5.416
Std	2.91231	32.82688	7.026427	7.3062	**0.0440304**	6.3649	4.1871	5.3637	21.9022	15.2892	5.8596	3.1959
Median	10.9203	299.0897	26.86388	64.5771	**0.1076456**	43.3136	8.8185	26.8639	18.7426	139.5265	29.8518	4.9023
Min	3.10826	241.4145	16.9143	52.4758	**0.0515707**	29.717	3.8941	13.9294	9.3234	109.9212	18.9075	2.508
Max	15.1387	403.9522	41.78825	77.3247	**0.221349**	51.9668	18.8062	33.8287	62.5372	171.2527	39.8013	8.8376
F_3												
Mean	1.13901	4.118388	0.008943	1.9129	0.3289801	15.9278	**2.748e−4**	0.0135	6.0126	7.9564	0.0083	305.1743
Std	1.16187	16.43281	0.0098425	1.0371	0.0937733	3.8231	**1.232e−4**	0.013	0.2969	1.434	0.002	5.0033
Median	0.856	1.117876	0.007396	1.5246	0.3234603	15.7031	**2.547e−4**	0.0089	5.9205	7.6865	0.0082	307.8109
Min	0.176166	1.076382	0	1.1524	0.1519639	11.1064	**1.628e−4**	0.0002	5.6253	5.3898	0.0054	299.4042
Max	4.67408	91.1243	0.0344575	5.2249	0.5169591	22.802	**4.068e−4**	0.052	6.3507	11.863	0.0137	308.308
F_4												
Mean	**1.85581**	4589.148	1876.432	283.8412	34.27486	51.2181	5.8172	52.7099	1452.9	728.1	58.7722	1414.8
Std	2.74797	832.2331	4697.424	113.1524	20.38742	33.3722	**1.384**	31.4375	472.7	193.8	34.2257	287
Median	**1.0669**	4397.42	161.5686	258.9218	27.09328	32.6838	5.8938	29.2173	1452.5	731.9	29.376	1509.4
Min	**0.15357**	3328.112	23.93475	143.4207	17.38066	29.277	1.1337	4.4676	887.2	432.3	24.8685	1005.6
Max	14.0421	7168.665	20078.28	534.3106	80.29335	111.4459	**9.4242**	110.5213	1963.2	1307.2	104.6797	1634.8

Table 4 Statistical data of the algorithms on mathematical benchmark functions (n = 100) (Best values are bolded)

F_1	LFABC	LFBBO	LFBSO	ABC	BBO	BSO	ELPSO	CPSO	HSO	GA	FSO	GSA
Mean	**5.407e-05**	604.1784	0.0039396	230.825	0.0219166	0.3696	6.035e-05	5.398e−4	76.0844	29.4081	2.211e−4	0.0672
Std	3.625e-05	28.2017	0.0015288	19.3176	0.0030417	0.0222	**1.860e−5**	2.639e−4	6.3151	4.6242	2.03e−5	0.0102
Median	**4.024e-05**	612.8889	0.0034889	237.4058	0.0219269	0.372	5.811e−5	4.095e−4	78.0859	31.2806	2.216e−4	0.0691
Min	**9.566e-06**	545.3276	0.0092637	198.6946	0.0170282	0.3406	4.297e−5	2.708e−4	69.0111	23.7734	2.001e−4	0.0562
Max	0.0001655	651.0005	0.0016971	249.1387	0.0302057	0.3937	**7.997e−5**	9.951e−4	81.1562	35.0934	2.524e−4	0.0763
F_2												
Mean	66.9981	1412.731	111.3497	753.3054	1.59653	164.6546	**5.5402**	78.782	415.5759	768.2887	214.1549	29.8488
Std	7.4067	53.35139	18.68541	12.2647	0.2806993	44.1605	**1.1612**	16.5476	171.3781	29.2261	30.192	1.9899
Median	68.2809	1416.113	114.4244	756.6776	1.602431	146.3873	**5.7829**	82.187	360.5646	775.294	218.9371	29.8488
Min	49.2414	1314.54	71.67504	735.1485	0.9935574	135.4258	**1.0448**	49.8618	300.2257	720.2649	169.1787	27.8589
Max	80.1129	1513.306	146.0594	767.0975	2.460157	230.4178	**9.793**	95.721	713.7281	797.1519	246.7955	31.8387
F_3												
Mean	1033.13	28.6495	0.0314067	817.3863	0.7864	83.5311	**0.2612**	0.2841	271.336	104.867	0.0152	1.1321
Std	82.7464	40.47719	0.0097468	63.1554	0.1172872	18.4563	0.1092	**0.041**	28.4434	11.0502	0.0011	0.3453
Median	1045.7	4.789451	0.0304533	844.5736	0.77688	84.5294	**0.2127**	0.2793	268.3095	108.7177	0.0151	0.6543
Min	753.727	3.902631	0.0152764	722.3432	0.60815	60.9781	**0.1846**	0.2347	244.5269	86.0552	0.0136	0.2126
Max	1169.06	95.00575	0.0538454	879.0898	1.17754	104.0877	0.3863	**0.3603**	301.1716	113.7519	0.0165	1.4532
F_4												
Mean	**4.35886**	29337.21	193.0249	1380800	137.3483	206.5988	8.739	209.9321	319290	26131	97.8001	93.0768
Std	3.07626	2067.502	42.41771	191900	46.48899	35.155	2.5665	83.3498	26200	3856	0.9119	**0.1221**
Median	**4.5148**	29758.55	192.0106	1278300	130.5773	215.7586	8.3733	225.3372	317150	25822	97.7481	93.0122
Min	**0.172849**	23389.19	118.0326	1252700	94.9541	156.385	3.7441	70.9597	294240	21330	96.4872	93.0005
Max	**9.89806**	33284.24	276.4468	1710000	255.0533	238.4931	18.0996	306.3208	34649	31699	98.7111	93.2176

Table 5 Search parameters of the optimization methods

Algorithm	428-member steel frame	1024-member space frame	693-member braced barrel vault
ABC	FS = 30, LFS(ABC/LFABC) = 150/100	FS = 30, LFS(ABC/LFABC) = 150/100	FS = 30, LFS(ABC/LFABC) = 150/100
BBO	NH(BBO/LFBBO) = 50/100, NED(BBO/LFBBO) = 2/5, m_{max} = 0.01	NH(BBO/LFBBO) = 50/100, NED(BBO/LFBBO) = 2/5, m_{max} = 0.05	NH(BBO/LFBBO) = 50/100, NED(BBO/LFBBO) = 2/5, m_{max} = 0.01
BSO	NI = 100, NCl(BSO/LFBSO) = 5/10, P_{S3} = 0.2, $P_{S4,1}$ = 0.8, $P_{S4,2}$ = 0.4	NI = 100, NCl(BSO/LFBSO) = 5/10, P_{S3} = 0.2, $P_{S4,1}$ = 0.8, $P_{S4,2}$ = 0.4	NI = 100, NCl(BSO/LFBSO) = 5/10, P_{S3} = 0.2, $P_{S4,1}$ = 0.8, $P_{S4,2}$ = 0.4

0.01. In LFBSO algorithm, the number of individuals, the number of cluster, the selection probability of the cluster center, randomly change probability of the cluster and, the probability of selecting one cluster are respectively taken as 5*n, 0.5*n, 0.8, 0.2 and 0.4. ABC and BSO algorithms are not reapplied on the benchmark functions. Their results are taken from a previous study (Jordehi 2015d). All tests are performed 30 times using different seed values and their statistical data are presented in Tables 2-4.

According to the test results, LFABC, LFBBO and LFBSO algorithms show satisfactory performances on the mathematical benchmark functions. In some functions, the LFABC and LFBSO algorithms show the best performance among all algorithms. In addition, the performances of the LFABC, LFBBO and LFBSO algorithms are generally better than their standard versions. Only LFBBO algorithm does not show better performance in some functions.

7. Design examples

In the study, three space steel skeleton structures are optimized in order to illustrate efficiency of LF strategy. First two structures are considered as steel space frames. The members of these structures are selected from the set of 272 W-sections starting from W100 × 19.3 to W1100 × 499 mm as given in LRFD-AISC (LRFD 2000). Space braced barrel vault is considered as last structure. Members of the structure are selected from the entire set of 37 standard circular hollow sections. Modulus of elasticity and shear modulus are taken as 200 GPa and 77 GPa respectively in all structures. Search parameters of the optimization method used in these structures are illustrated in Table 5.

Table 6 Member grouping of the first design example

Story	Side beam	Inner beam	Corner column	Side column	Inner column
1	1	2	9	10	11
2	3	4	12	13	14
3	5	6	15	16	17
4	7	8	18	19	20

Fig. 2 3D view of the first design example

Table 7 Load details and displacement limitations of the first and second examples

Load type	428 member frame	1024 member frame
Dead load	2.88 kN/m^2	2.88 kN/m^2
Live load	2.39 kN/m^2	2.39 kN/m^2
Snow load	0.755 kN/m^2	0.755 kN/m^2
Wind speed	38 m/s	38 m/s
Top story D.	3.5 cm	3.5 cm
Inter-story D.	0.875 cm	0.875 cm
Def. limit	2 cm	2 cm

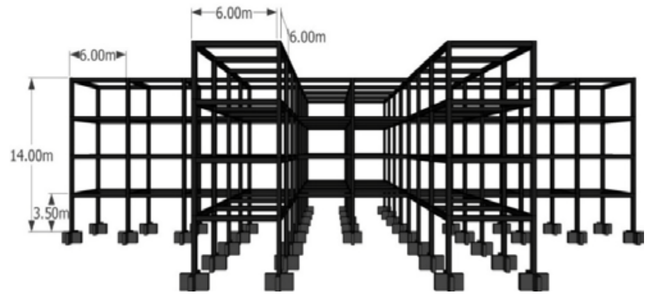


Fig. 3 Side view of the first design example

Table 8 Design details of the best solutions for the first example (NA: Not Available)

#		ABC	LFABC	BBO	LFBBO	BSO	LFBSO
1	Beam	W310×38.7	W410×46.1	W360×32.9	W310×28.3	W310×32.7	W310×23.8
2	Beam	W360×32.9	W310×28.3	W250×32.7	W310×38.7	W200×26.6	W310×21
3	Beam	W460×52	W360×44	W460×52	W360×39	W310×32.7	W360×44
4	Beam	W460×52	W460×52	W310×32.7	W410×38.8	W250×49.1	W360×32.9
5	Beam	W310×32.7	W410×60	W530×66	W460×52	W610×82	W410×46.1
6	Beam	W310×32.7	W410×38.8	W460×52	W530×92	W250×73	W410×38.8
7	Beam	W310×38.7	W360×32.9	W360×32.9	W460×60	W460×52	W360×32.9
8	Beam	W360×39	W410×60	W460×52	W460×52	W610×113	W250×32.7
9	Column	W200×46.1	W460×144	W410×53	W310×97	W250×49.1	W310×79
10	Column	W200×46.1	W310×38.7	W250×49.1	W200×52	W310×86	W310×86
11	Column	W200×46.1	W200×46.1	W200×46.1	W310×38.7	W200×71	W360×91
12	Column	W840×210	W460×144	W410×100	W310×97	W310×107	W360×134
13	Column	W460×74	W310×143	W250×80	W360×72	W310×86	W310×107
14	Column	W690×140	W200×46.1	W360×134	W410×100	W250×73	W460×97
15	Column	W1000×321	W460×144	W460×113	W310×97	W310×107	W360×216
16	Column	W920×201	W310×143	W310×97	W460×89	W310×86	W310×107
17	Column	W760×147	W360×147	W360×147	W410×100	W250×73	W760×196
18	Column	W1100×390	W760×173	W920×201	W460×113	W310×107	W360×216
19	Column	W920×201	W690×217	W840×193	W460×89	W310×86	W690×125
20	Column	W760×147	W360×216	W530×150	W410×100	W250×73	W920×342
Max. strength ratio		1	0.883	0.978	0.902	0.988	0.998
Top drift (cm)		2.91	3.01	2.867	3.266	3.325	2.933
Inter story drift (cm)		0.875	0.535	0.875	0.872	0.866	0.87
Max. deflection (cm)		0.49	0.512	NA	0.246	0.385	0.234
Maximum iteration		50000	50000	50000	50000	50000	50000
Weight (kN)		1512.11	1481.73	1332.29	1239.21	1354.02	1326.74

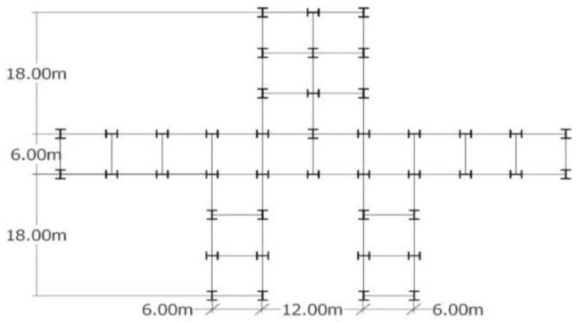


Fig. 4 Plan view of the first design example

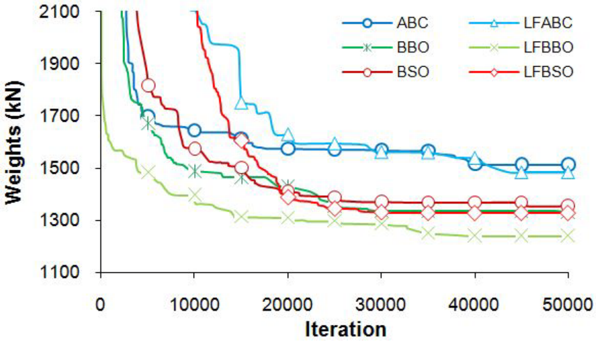


Fig. 5 Search histories of best designs for the first example

7.1 Four-story, 428-member steel frame

428 member 3-D frame which is previously used in the literature (Aydoğdu and Akın 2014, Akın and Aydoğdu 2015, Aydoğdu *et al*. 2016, Çarbaş 2016) is considered as the first design example in the study. 3-D, side and plan views of the frame are shown in Figs. 2-4, respectively. In the example problem, the frame members are grouped into 20 independent design groups which are illustrated in Table 6. Dead, live, snow and wind loads are considered for the design of the structure. The design loads and load combinations are computed from the ASCE 7-05 (ASCE 7-05 2005). The load combinations of the study are described as: 1.2DL + 1.6LL + 0.5SL, 1.2DL + 0.5LL + 1.6SL, 1.2DL + 1.6WL + LL + 0.5SL. The load details and displacement limitations are illustrated in Table 8.

It is clearly illustrated in the table that the lightest weight is obtained as 1239.21 kN by using the LFBBO algorithm. This weight is 7.06%, 7.51%, 9.26%, 19.57% and 22.02% lighter than the optimum weights of LFBSO, BBO, BSO, LFABC and ABC algorithms respectively. In addition, when LF distribution is taken into account, 2.05%, 2.06% and 7.51% lighter designs are obtained for ABC, BSO and BBO algorithms respectively. The design histories of these algorithms for the best solutions are also plotted in Fig. 5.

From Table 8, it can be concluded that while the inter-story drift constraint is active in the optimum designs, the top-story sway limitation is relatively not active. Additionally, it is noted from this table that the strength limitations are dominant in the design problem. In the optimum frames, the strength ratios of some members are very close to their upper bound of 1.0. Hence, in the optimum results, it is the strength constraints that govern the design.

Another interesting result derived from Fig. 5 is that the optimum design algorithms with or without the effect of LF perform similar convergence rates except LFBBO. LFBBO has better convergence than the others. For this example, the deflection values observed for all optimum designs are very low.

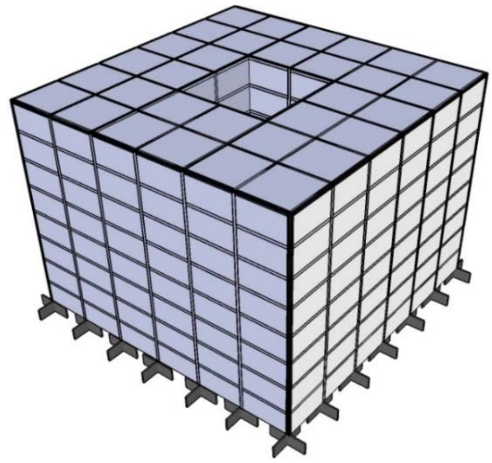


Fig. 6 3-D view of the second design example

Table 9 Member grouping of the second design example

Story	Side beam	Inner beam	Corner column	Side column	Inner column
8	1	2	17	18	19
7	3	4	20	21	22
6	5	6	23	24	25
5	7	8	26	27	28
4	9	10	29	30	31
3	11	12	32	33	34
2	13	14	35	36	37
1	15	16	38	39	40

7.2 Eight-story, 1024-member space frame

In the second design example taken from previous studies (Aydoğdu *et al*. 2012a, b, 2016, Aydoğdu and Akın 2014), the eight-story steel space frame has 1024 members and 384 joints collected into 40 independent design groups. 3-D, plan and side views of the frame are shown in Figs. 6-8, respectively. The member grouping of the frame is given Table 7. The frame members are selected from the set of 272 W-sections starting from W100 × 19.3 to W1100 × 499 mm as given in LRFD-AISC (LRFD 2000). The load combinations of the study are described as: 1.2DL + 1.6LL + 0.5SL, 1.2DL + 0.5LL + 1.6SL, 1.2DL + 1.6WXL + LL + 0.5SL and 1.2DL + 1.6WYL + LL + 0.5SL. The design loads, basic wind speed, drift and deflection limits of the frame are computed according to ASCE 7-05 (ASCE 7-05 2005) and Ad Hoc Committee on Serviceability, are illustrated in Table 7.

Table 10 Design details of the best solutions for the second example (NA: Not Available)

#	Group Type	ABC	LFABC	BBO	LFBBO	BSO	LFBSO
1	Beam	W360×32.9	W200×26.6	W310×21	W200×46.1	W200×26.6	W410×114
2	Beam	W250×32.7	W150×37.1	W310×21	W530×150	W200×26.6	W410×114
3	Beam	W530×92	W410×60	W410×53	W250×73	W410×60	W410×114
4	Beam	W250×58	W460×60	W250×49.1	W530×150	W360×51	W410×114
5	Beam	W360×101	W360×79	W610×101	W200×46.1	W250×44.8	W410×114
6	Beam	W200×71	W410×100	W250×58	W250×49.1	W310×67	W410×114
7	Beam	W530×92	W310×32.7	W410×100	W200×59	W530×74	W410×114
8	Beam	W530×66	W310×32.7	W250×80	W530×150	W310×79	W410×114
9	Beam	W690×140	W530×85	W610×155	W310×79	W530×92	W410×114
10	Beam	W530×66	W250×32.7	W530×165	W250×80	W460×89	W410×114
11	Beam	W760×134	W250×131	W690×125	W360×91	W610×92	W410×114
12	Beam	W310×28.3	W360×57.8	W310×107	W530×150	W460×89	W410×114
13	Beam	W840×176	W760×196	W610×125	W610×101	W530×92	W410×114
14	Beam	W410×60	W530×66	W460×52	W530×150	W610×101	W410×114
15	Beam	W760×173	W460×97	W840×176	W760×134	W530×92	W410×114
16	Beam	W410×38.8	W310×32.7	W310×28.3	W530×150	W530×101	W410×114
17	Column	W360×44	W310×74	W310×107	W310×158	W610×92	W410×114
18	Column	W410×100	W530×92	W360×196	W530×150	W360×101	W410×114
19	Column	W200×52	W200×35.9	W200×41.7	W530×150	W610×92	W410×114
20	Column	W460×128	W360×287	W310×117	W310×158	W690×217	W410×114
21	Column	W610×195	W840×210	W360×196	W530×150	W690×217	W410×114
22	Column	W250×58	W200×35.9	W250×58	W530×150	W690×217	W410×114
23	Column	W1000×321	W360×287	W360×162	W310×158	W760×314	W410×114
24	Column	W760×284	W1000×296	W360×237	W530×150	W760×314	W410×114
25	Column	W360×162	W530×165	W250×67	W530×150	W760×314	W410×114
26	Column	W1000×321	W310×375	W360×162	W310×158	W1000×415	W410×114
27	Column	W840×299	W1000×412	W360×347	W530×150	W1100×390	W410×114
28	Column	W360×262	W1000×258	W310×97	W530×150	W1100×390	W410×114
29	Column	W1000×443	W310×375	W610×174	W310×158	W1100×433	W410×114
30	Column	W1000×321	W1000×412	W1100×390	W530×150	W1100×433	W840×226
31	Column	W360×287	W1000×258	W530×150	W530×150	W1100×433	W410×114
32	Column	W1000×477	W310×375	W760×314	W310×158	W1100×499	W410×114
33	Column	W1100×433	W1000×412	W1100×499	W530×150	W1100×499	W1100×499
34	Column	W1000×321	W1000×258	W530×150	W530×150	W1100×499	W410×114
35	Column	W1000×477	W610×415	W920×446	W310×158	W1100×499	W410×114
36	Column	W1100×499	W1000×412	W1100×499	W530×150	W1100×499	W1100×499
37	Column	W1000×321	W1000×258	W920×201	W530×150	W1100×499	W410×114
38	Column	W1000×477	W610×415	W1000×443	W310×158	W1100×499	W410×114
39	Column	W1100×499	W1000×412	W1100×499	W530×150	W1100×499	W1100×499
40	Column	W1000×321	W1000×258	W1000×258	W530×150	W1100×499	W840×226
Max. strenght ratio		0.986	0.989	1	0.91	0.998	0.968
Top drift (cm)		6.577	6.239	6.508	5.095	4.986	4.905
Inter storey drift (cm)		0.874	0.871	0.875	0.874	0.714	0.827
Max. deflection (cm)		N.A	N.A	N.A	0.29	0.115	0.618
Maximum iteration		75000	75000	75000	75000	75000	75000
Weight (kN)		7210.12	7689.51	6462.79	6092.91	7652.84	6793.57

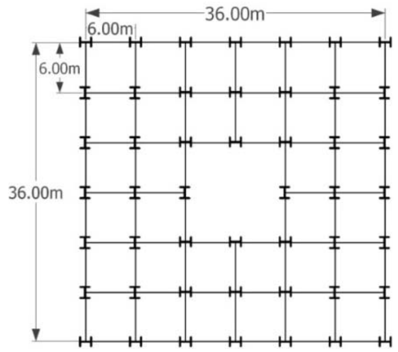


Fig. 7 Plan view of the second design example

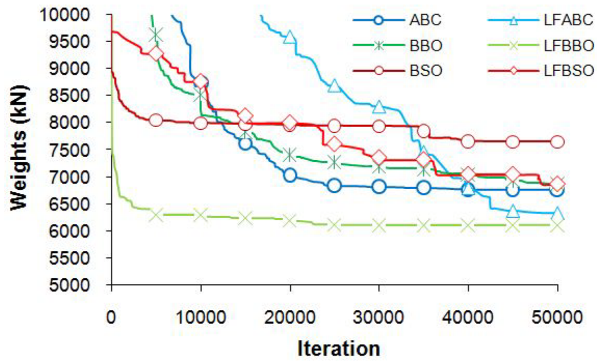


Fig. 9 Search histories of best designs for the second example

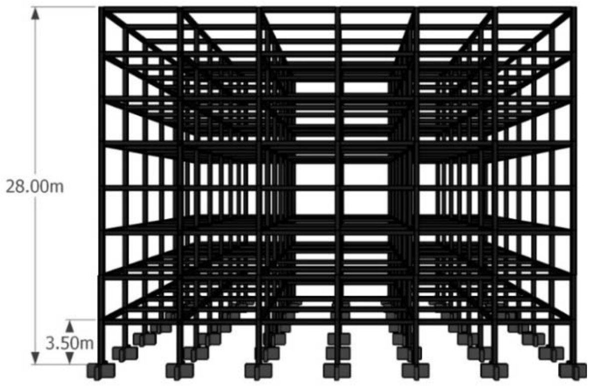


Fig. 8 Side view of the second design example

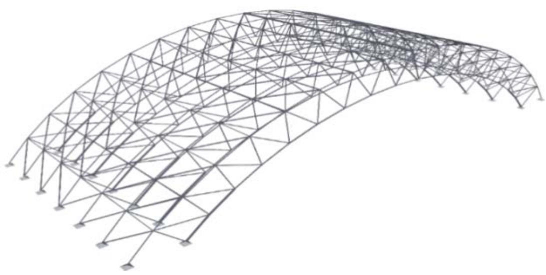


Fig. 10 3D view of the third design example

The example is solved by the LFBBO, BSO and LFBSO algorithms. The obtained results are compared to results of ABC, BBO and LFABC algorithms achieved by the authors formerly (Aydoğdu and Akın 2014, Aydoğdu *et al*. 2016, Çarbaş 2016). The lightest weight, the maximum strength ratio, maximum displacements and W-section designations of the optimum designs are given in the Table 10. It is clearly illustrated in the table that the lightest weight is obtained as 6092.91 kN by using the LFBBO algorithm. This weight is 0.83%, 6.07%, 10.97%, 11.5%, and 25.6% lighter than the optimum weights of LFABC, BBO, ABC, BSO, and LFBSO algorithms respectively. In addition, when LF distribution is taken into account, 6.07%, 10.17% and 14.1% lighter designs are obtained for BBO, ABC and BSO algorithms respectively. The design histories of these algorithms for the best solutions are also plotted in Fig. 9.

The eight-story, 1024-member steel skeletal frame is the most challenging design example of this study with the greatest number of design variables. Both the strength ratio constraints and the drift constraints are active for this example. Both strength and serviceability constraints are dominant in the optimum design as shown in Table 10. From optimal design attained by BBO algorithm, the strength ratios of some members are at the upper bound of 1.0, while attained by other algorithms are close to the

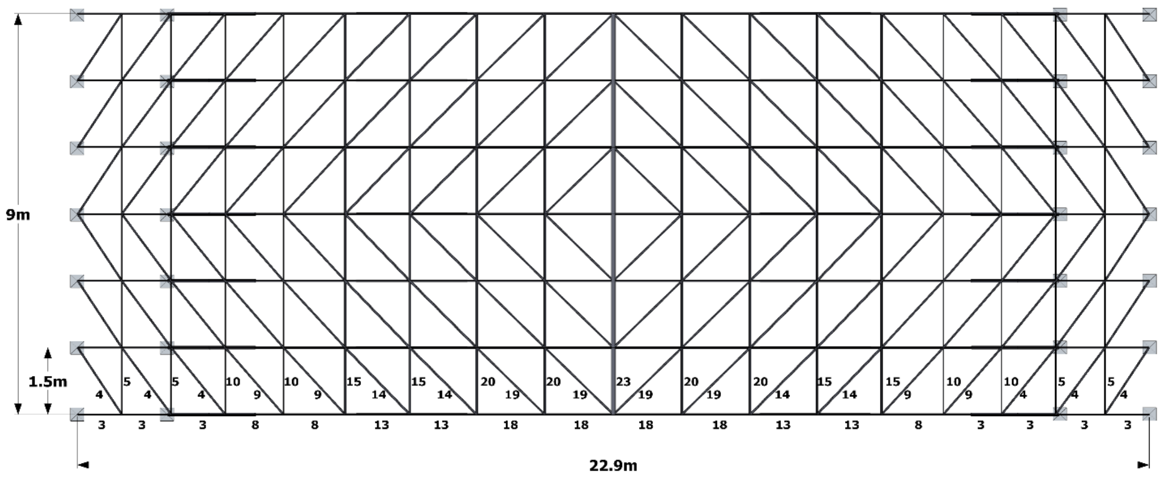


Fig. 11 Plan view of the third design example

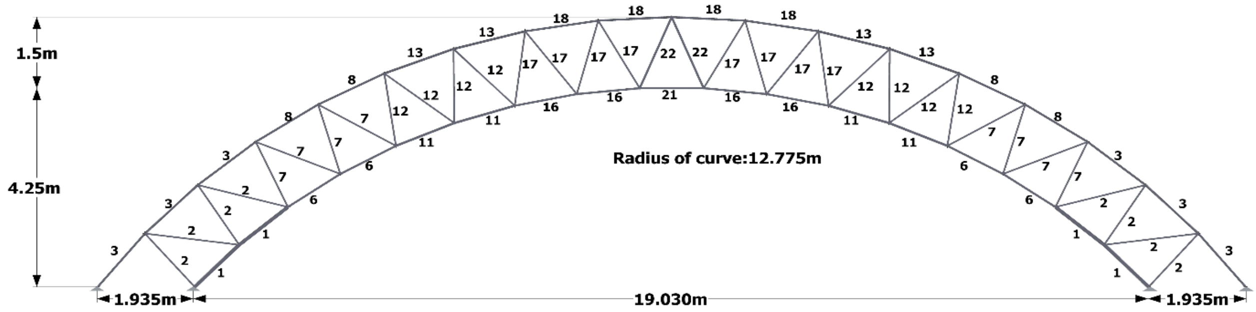


Fig. 12 Side view of the third design example

Table 11 Design details of the best solutions for the third example

	ABC	LFABC	BSO	LFBSO	BBO	LFBBO
1	P2.5	PX1.25	PX1.25	P2.5	PX1.25	P2.5
2	P1	P1	P1	P1	P1	P1
3	P.75	P1	P1	P.75	P1	P.75
4	P1	P1	P1	P1	P1	P1
5	P.75	P.75	P.75	P.75	P.75	P.75
6	PX1.25	P1.25	P1.25	PX1.25	P1.25	P1.5
7	P1	P1	P1	P1	P1	P1
8	P1	PX1	PX1	P1	PX1	P1
9	P1	P1	P1	P1	P1	P1
10	P.75	P1	P.75	P.75	P1	P.75
11	PX1	PX1.25	P1	P1.25	P1	P1.25
12	P1	P1	P1	P1	P1	P1
13	P1	PX1	P2.5	P1	P2.5	P1
14	P1	P1.25	P1.25	P1	P1.25	P1
15	P3.5	P.75	P.75	P.75	P.75	P.75
16	PX1	P1	P1	PX1	P1	PX1
17	P1	PX1	P.75	P1	P.75	P1
18	P1	PX1	P1.25	P1	P1.25	P1
19	P1	P1.5	P1.25	P1	P1.5	P1
20	P.75	P.75	P1.25	P.75	P1.25	P.75
21	P1	PX.75	PX.75	PX.75	PX.75	PX1
22	P.75	P1.5	P.75	P.75	P.75	P.75
23	P2	PX1.25	P.75	PX3	P3	PX1.25
Max. Str. R.	0.995	0.989	0.934	0.989	0.937	0.999
Max. Def. (cm)	0.753	0.813	0.839	0.741	0.833	0.771
Max. Iter.	30000	30000	30000	30000	30000	30000
Weight (kN)	33.15	31.28	30.66	29.86	32.11	28.85

upper bound of 1.0. The inter-story drifts in all optimum designs have nearly an upper bound of 0.875. The top-story drifts located at the joints on the top story of the steel frame do not have values close their upper bounds of 7.0 for any optimum design. LFBBO shows rapid convergence to the optimum design compared to the other algorithms, Fig. 8. The deflection values observed for all optimum designs are also very low.

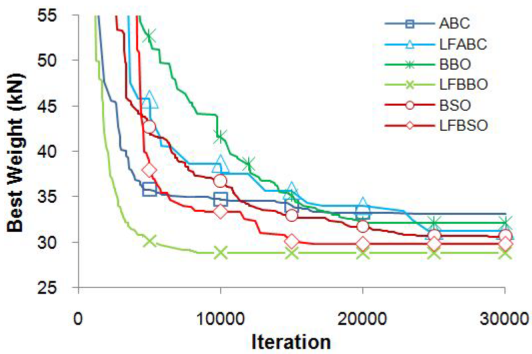


Fig. 13 Search histories of best designs for the third example

7.3 693-member braced barrel vault

The third design example taken from previous studies (Hasançebi and Çarbaş 2011, Hasancebi and Azad 2014) is the spatial braced barrel vault. The structure has 693 members and 259 joints collected into 23 independent design groups. 3-D, plan and views of the structure are shown in Figs. 9, 10 and 11 respectively. Member grouping of the structure is illustrated in these figs. as well. Structure is exposed to both dead (35 kg/m^2) and wind loads (positive wind load: 160 kg/m^2; negative wind load 240 kg/m^2). Two load combinations are considered in the example which is 1.5D+1.5W+ and 1.5D+1.5W–. The displacements of all joints in all directions are limited to 6.36 cm. The strength and stability requirements of steel members are imposed according to AISC-LRFD (1999). Sap2000 Open Application Programming Interface (OAPI) is used to analyze and design of the structure.

The structure is optimized using the ABC, LFABC, BBO, LFBBO, BSO and LFBSO algorithms. The lightest weight, the maximum strength ratio, maximum displacements and W-section designations of the optimum designs are given in the Table 11. It is clearly illustrated in the table that the lightest weight is obtained as 28.85 kN by using the LFBBO algorithm. This weight is 3.49%, 6.45%, 8.42%, 11.31% and 14.9% lighter than the optimum weights of the LFBSO, BSO, LFABC, BBO and ABC algorithms respectively. After usage of LF distribution, 2.67%, 5.97%, 11.31% lighter designs are obtained for the BSO, ABC and BBO algorithms respectively. The design histories of these algorithms for the best solutions are also plotted in Fig. 13.

8. Sensitivity analysis of the control parameters

Metaheuristic techniques use two main strategies to search for the optimum solution. These are diversification and intensification. Efficient metaheuristic method should construct dynamic balance between diversification and intensification. Detecting suitable control parameters of the metaheuristic algorithms is one of the most preferred techniques to construct balance. However, the suitable control parameters depend on the optimization problem structure. Therefore, sensitivity analysis can be required for different types of optimization problems. Structural optimization problems have different structures than the standard benchmark mathematical optimization problems. Hence, control parameters of metaheuristics should only be obtained from sensitivity analysis for structural optimization problems. Control parameters of the ABC, LFABC, BBO and LFBBO algorithms for structural optimization problems (frame and retaining wall structures) are determined using sensitivity analysis or test problems in previous studies (Aydoğdu *et al*. 2016, Çarbaş 2016, Aydoğdu 2017). In this study, the control parameters of these algorithms are adopted from the previous studies.

Computation time of single optimization procedure for each design example exceeds one day. Therefore, design examples are not reasonable for sensitivity analysis. Hence, 105 member space frame is preferred for sensitivity analysis of BSO and LFBSO algorithms which are previously used in many studies (Aydoğdu and Saka 2009, Aydoğdu 2010, Saka *et al*. 2011, Akın and Aydoğdu 2015). The frame has 54 joints and 105 members that are grouped into 11 independent design variables. 3-D and plan views of the structure are given in Figs. 14-15. Both gravity and lateral loads which are computed per ASCE 7-10 (~~Committee~~ 2010) applied to the frame. The design dead and live loads are taken as 2.88 kN/m^2 and 2.39 kN/m^2. Ground snow load is considered to be 0.755 kN/m^2 and a basic wind speed is 105 mph (65 m/s). The unfactored distributed gravity loads on the beams of the roof and floors are tabulated in Table 12. The following load combinations are considered in the design of the frame according to code specification. 1.2D + 1.6L + 0.5S, 1.2D + 0.5L + 1.6S, 1.2D + 1.6W + 0.5L + 0.5S where D is the dead load, L is the live load, S is the snow load and W is the wind load. The structure is tested forty-two times by considering different values for NI (25, 50 and 100), NCl (0.05*NI and 0.1*NI) and $P_{S4,1}$ (0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9). Ps_3 and $P_{S4,1}$ parameters are not included in the sensitivity analysis since these parameters are not effective for sensitivity analysis. Therefore, values Ps_3 and $\mathrm{P}_{S4,1}$ parameters are determined according to experimental studies in the literature (Shi 2011, 2014). In each test, the frame is optimized ten times using different seed values. Mean values of these tests are illustrated in Tables 13-14. According to the tables, most suitable control parameters are bolded in the tables.

9. Conclusions

In this study, the effect of the LF strategy on stochastic search techniques is investigated in the optimum design of steel skeleton structures. For this purpose, three real size steel structures are optimized using three stochastic search techniques (ABC, BBO and BSO) and their enhanced versions which make use of the LF strategy. It is observed from the results obtained from the design examples that make use of the LF strategy yields 2.5%-14.1% lighter optimum designs. The effect of the LF strategy even becomes very effective in the case of BBO and LFBBO algorithms in the design of first and third structures. The difference between BBO and LFBBO methods is 7.51% in the first problem and 11.31% in the third. The effect of LF strategy is less effective for ABC and BSO methods in these design examples. The differences in these methods vary from 2.05% to 5.97%. For the second design example, which contains more design variables and constraint functions than the first and third problems, 6.07%-14.1% lighter optimum designs are obtained when LF strategy is considered. In this example, the largest weight difference is perceived in the BSO results; whereas, the smallest weight difference is observed for the BBO method. Based on comparison of the results, the LF strategy considerably increases the robustness and efficiencies of the stochastic search algorithms considered in this study. For the BBO algorithm, the LF strategy is more effective in the small-scale problem. On the contrary, the LF strategy is more effective in the large-scale problem for the BSO algorithm. It can be concluded that the LF strategy can have a notable effect in enhancing the performance of the stochastic search techniques in the optimum design of steel skeleton structures. Furthermore, the LFBBO algorithm showed the best performance among all design examples.

Although, LF strategy can have a notable effect in enhancing the performance of the stochastic search techniques for the optimum design of the steel skeleton structures, alternative strategies like chaotic operators, whose efficiencies are proved for different optimization problems, are also available in the literature (Talatahari *et al*. 2012, Gandomi and Yang 2014, Jordehi 2014, 2015b, c, Kaveh *et al*. 2014) and these alternative strategies could be compared in future studies to LF.

References

Akın, A. and Aydoğdu, İ. (2015), “Optimum design of steel space frames by hybrid teaching-learning based optimization and harmony search algorithms”, *Int. J. Mech. Aerosp. Indust. Mechatron. Manuf. Eng.*, **9**(7), 1367-1374.

Al-Temeemy, A.A., Spencer, J. and Ralph, J. (2010),“Levy flights for improved ladar scanning”, *Proceedings of 2010 IEEE International Conference on Imaging Systems and Techniques*, Thessaloniki, Greece, July.

Artar, M. and Daloglu, A.T. (2015a), “The optimization of multi-storey composite steel frames with genetic algorithm including dynamic constraints”, *Tek Dergi*. **26**(2), 7077-7098.

Artar, M. and Daloglu, A.T. (2015b), “Optimum design of steel frames with semi-rigid connections and composite beams”, *Struct. Eng. Mech., Int. J.*, **55**(2), 299-313.

Artar, M. and Daloglu, A.T. (2015c), “Optimum design of steel space frames with composite beams using genetic algorithm”,, *Steel Compos. Struct., Int.J.*, **19**(2), 503-519.

ASCE 7-05 (2005), *Minimum Design Loads for Buildings and Other Structures*, American Society of Civil Engineers, Reston,

VA, USA, ABD.

Aydoğdu, İ. (2010), *Optimum Design of 3-D Irregular Steel Frames Using Ant Colony Optimization and Harmony Search Algorithms*, Ph.D. Thesis; Middle East Technical University, August, Ankara, Turkey.

Aydogdu, I. (2017), "Cost optimization of reinforced concrete cantilever retaining walls under seismic loading using a biogeography-based optimization algorithm with Levy flights", *Eng. Optimiz.*, **49**(3), 381-400.

Aydoğdu, İ. and Akın, A. (2014), "Teaching and learning-based optimization algorithm for optimum design of steel buildings", *Comput. Civil Build. Eng.*, 2167-2175.

Aydogdu, I. and Saka, M.P. (2009), "Ant colony optimization of irregular steel frames including effect of warping", *Proceedings of the 12th International Conference on Civil, Structural and Environmental Engineering Computing*, Madeira, Portugal, September.

Aydoğdu, İ., Akın, A. and Saka, M. (2012a), *Discrete Design Optimization of Space Steel Frames using the Adaptive Firefly Algorithm*, Civil Comp, Dubrovnik, Crotia.

Aydoğdu, İ., Akın, A. and Saka, M. (2012b), *Optimum Design of Steel Space Frames by Artificial Bee Colony Algorithm,* Ankara, Turkey.

Aydoğdu, İ., Akın, A. and Saka, M. (2016), "Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution", *Adv. Eng. Software,* **92**, 1-14.

Ben Hadj-Alouane, A. and Bean, J.C. (1997), "A genetic algorithm for the multiple-choice integer program", *Operat. Res.*, **45**(1), 92-101.

Bhattacharya, A. and Chattopadhyay, P.K. (2010), "Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch", *IEEE Transactions on Power Systems*, **25**(4), 1955-1964.

Çarbaş, S. (2016), "Optimum structural design of spatial steel frames via biogeography-based optimization", *Neural Comput. Appl.*, 1-15.

Carbas, S. and Aydoğdu, I. (2017), "Utilization of harmony search algorithm in optimal structural design of cold-formed steel structures", *Proceedings of International Conference on Harmony Search Algorithm*, Bilbao, Spain, February.

Carbas, S., Aydoğdu, I. and Saka, M.P. (2009), "Optimum design of steel frames using adaptive harmony search method to LRFD-AISC", *Proceedings of the 8th World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, June.

Carbas, S., Aydoğdu, I. and Saka, M.P. (2013), "A comparative study of three metaheuristics for optimum design of engineering structures", *Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, FL, USA, May.

Chen, Y. (2010), "Research and simulation on Levy flight model for DTN", *Proceedings of the 3rd International Congress on Image and Signal Processing* (*CISP*), Yantai, China, October.

Cheng, S., Qin, Q., Chen, J., Wang, G.-G. and Shi, Y. (2016), "Brain storm optimization in objective space algorithm for multimodal optimization problems", *Advances in Swarm Intelligence*: *Proceedings of the 7th International Conference, ICSI*, Bali, Indonesia, June.

Coello, C.A.C. (2002), "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art", *Comput. Method. Appl. Mech. Eng.*, **191**(11), 1245-1287.

Coello, C.A.C. and Carlos, A. (1999), "A survey of constraint handling techniques used with evolutionary algorithms", *Lania-RI-99-04*, *Laboratorio Nacional de Informática Avanzada*.

Degertekin, S. (2012), "Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm", *Steel Compos. Struct., Int. J.*, **12**(6), 505-522.

Digalakis, J.G. and Margaritis, K.G. (2002), "An experimental study of benchmarking functions for genetic algorithms", *Int. J. Comput. Math.*, **79**(4), 403-416.

Dorigo, M. (1992), *Optimization, Learning and Natural Algorithms*, Politecnicodi Milano, Milano, Italy.

Eberhart, R.C. and Kennedy, J. (1995), "A new optimizer using particle swarm theory", *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, October.

Ellingwood, B. (1986), "Structural Serviceability: A Critical Appraisal and Research Needs", *J. Struct. Eng.*, **112**(12), 2646-2664.

Fourie, P. and Groenwold, A.A. (2002), "The particle swarm optimization algorithm in size and shape optimization", *Struct. Multidiscipl. Optimiz.*, **23**(4), 259-267.

Gandomi, A.H. and Yang, X.-S. (2014), "Chaotic bat algorithm", *J. Computat. Sci.*, **5**(2), 224-232.

Geem, Z.W., Kim, J.H. and Loganathan, G. (2001), "A new heuristic optimization algorithm: harmony search", *Simulation*, **76**(2), 60-68.

Gong, W., Cai, Z., Ling, C.X. and Li, H. (2010), "A real-coded biogeography-based optimization with mutation", *Appl. Math. Computat.*, **216**(9), 2749-2758.

Hadidi, A., Azad, S.K. and Azad, S.K. (2010), "Structural optimization using artificial bee colony algorithm", *Proceedings of the 2nd International Conference on Engineering Optimization*, Lisbon, Portugal, September.

Hasancebi, O. and Azad, S.K. (2014), "Discrete size optimization of steel trusses using a refined big bang-big crunch algorithm", *Eng. Optimiz.*, **46**(1), 61-83.

Hasançebi, O. and Çarbaş, S. (2011), "Ant colony search method in practical structural optimization", *Int. J. Optimiz. Civil Eng.*, **1**(1), 91-105.

Homaifar, A., Qi, C.X. and Lai, S.H. (1994), "Constrained optimization via genetic algorithms", *Simulation*, **62**(4), 242-253.

Jalili, S., Hosseinzadeh, Y. and Kaveh, A. (2014), "Chaotic biogeography algorithm for size and shape optimization of truss structures with frequency constraints", *Periodica Polytechnica. Civil Engineering*, **58**(4), 397.

Joines, J.A. and Houck, C.R. (1994), "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's", *Proceedings of the 1st IEEE Conference on Evolutionary Computation IEEE World Congress on Computational Intelligence*, New York, NY, USA, June.

Jordehi, A.R. (2014), "A chaotic-based big bang–big crunch algorithm for solving global optimisation problems", *Neural Comput. Appl.*, **25**(6), 1329-1335.

Jordehi, A.R. (2015a), "Brainstorm optimisation algorithm (BSOA): An efficient algorithm for finding optimal location and setting of FACTS devices in electric power systems", *Int. J. Electric. Power Energy Syst.*, **69**, 48-57.

Jordehi, A.R. (2015b), "A chaotic artificial immune system optimisation algorithm for solving global continuous optimisation problems", *Neural Comput. Appl.*, **26**(4), 827-833.

Jordehi, A.R. (2015c), "Chaotic bat swarm optimisation (CBSO)", *Appl. Soft Comput.*, **26**, 523-530.

Jordehi, A.R. (2015d), "Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems", *Appl. Soft Comput.*, **26**, 401-417.

Jordehi, A.R. (2015e), "A review on constraint handling strategies in particle swarm optimisation", *Neural Comput. Appl.*, **26**(6), 1265-1275.

Karaboga, D. (2005), *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report-tr06; Erciyes

University, Engineering Faculty, Computer Engineering Department.

Karaboga, D. and Basturk, B. (2007), “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, *J. Global Optimiz.*, **39**(3), 459-471.

Karaboga, D. and Basturk, B. (2008), “On the performance of artificial bee colony (ABC) algorithm”, *Appl. Soft Comput.*, **8**(1), 687-697.

Kaveh, A., Sheikholeslami, R., Talatahari, S. and Keshvari-Ilkhichi, M. (2014), “Chaotic swarming of particles: a new method for size optimization of truss structures”, *Adv. Eng. Softw.*, **67**, 136-147.

Lee, K.S. and Geem, Z.W. (2004), “A new structural optimization method based on the harmony search algorithm”, *Comput. Struct.*, **82**(9), 781-798.

Lenin, K., Ravindhranath Reddy, B. and Surya Kalavathi, M. (2014), “Brain storm optimization algorithm for solving optimal reactive power dispatch problem”, *Int. J. Res. Electron. Commun. Technol.*, **1**(3).

Levy, P. (1939), “L’addition des variables aléatoires définies sur une circonférence”, *Bulletin de la Société Mathématique de France*, **67**, 1-41.

Li, J.N. and Duan, H.B. (2015), “Simplified brain storm optimization approach to control parameter optimization in F/A-18 automatic carrier landing system”, *Aerosp. Sci. Technol.*, **42**, 187-195.

LRFD (2000), Load & Resistance Factor Design Specification; *American Institute of Steel Construction*, Chicago, IL, USA.

Michalewicz, Z. and Nazhiyath, G. (1995), “Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints”, *Proceedings of IEEE International Conference on Evolutionary Computation*, Perth, Australia, November-December.

Pereyra, M.A. and Batatia, H. (2010), “A Levy flight model for ultrasound in skin tissues”, *Proceedings of 2010 IEEE International Ultrasonics Symposium*, San Diego, CA, USA, October.

Perez, R. and Behdinan, K. (2007), “Particle swarm approach for structural design optimization”, *Comput. Struct.*, **85**(19), 1579-1588.

Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2009), “GSA: A gravitational search algorithm”, *Inform. Sci.*, **179**(13), 2232-2248.

Roy, P.K., Ghoshal, S.P. and Thakur, S.S. (2011), “Optimal reactive power dispatch considering flexible AC transmission system devices using biogeography-based optimization”, *Electric Power Components Syst.*, **39**(8), 733-750.

Saka, M.P., Aydoğdu, I. and Hasancebi, O. (2010), “Evaluation of recent improvements in harmony search based structural optimization algorithms”, *Proceedings of the 10th International Conference on Computational Structures Technology*, Valencia, Spain, September.

Saka, M.P., Aydoğdu, I., Hasancebi, O. and Geem, Z.W. (2011), *Harmony Search Algorithms in Structural Engineering*, Springer.

Saka, M., Carbas, S., Aydoğdu, I., Akin, A. and Geem, Z. (2015), *Comparative Study on Recent Metaheuristic Algorithms in Design Optimization of Cold-Formed Steel Structures*, Springer, Cham, Switzerland.

Salcedo-Sanz, S. (2009), “A survey of repair methods used as constraint handling techniques in evolutionary algorithms”, *Comput. Sci. Review*, **3**(3), 175-192.

Shi, Y. (2011), *Brain Storm Optimization Algorithm*, Springer.

Shi, Y. (2014), “An optimization algorithm based on brainstorming process”, *Emerg. Res. Swarm Intel. Algorithm Optimiz.*, **1**.

Shi, Y. and Eberhart, R. (1998), “A modified particle swarm optimizer”, *Proceedings of the 1998 IEEE World Congress on Computational Intelligence*; *IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, May.

Simon, D. (2008), “Biogeography-based optimization”, *IEEE Transact. Evolution. Computat.*, **12**(6), 702-713.

Sonmez, M. (2011), “Discrete optimum design of truss structures using artificial bee colony algorithm”, *Struct. Multidiscipl. Optimiz.*, **43**(1), 85-97.

Surry, P.D., Radcliffe, N.J. and Boyd, I.D. (1995), “A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA Method”, *AISB Workshop on Evolutionary Computing*.

Talatahari, S., Kaveh, A. and Sheikholeslami, R. (2012), “Chaotic imperialist competitive algorithm for optimum design of truss structures”, *Struct. Multidiscipl. Optimiz.*, **46**(3), 355-367.

Terdik, G. and Gyires, T. (2009), “Lévy flights and fractal modeling of internet traffic”, *IEEE/ACM Transactions on Networking*, **17**(1), 120-129.

Wang, J.S., Shen, N.N., Ren, X.D. and Liu, G.N. (2015), “RBF neural network soft-sensor modeling of Rotary Kiln Pellet quality indices optimized by biogeography-based optimization algorithm”, *J. Chem. Eng. Jpn*, **48**(1), 7-15.

Yang, X.-S. (2010), *Nature-Inspired Metaheuristic Algorithms*, Luniver Press.

Yang, X.-S., Bekdaş, G. and Nigdeli, S.M. (2016), *Review and Applications of Metaheuristic Algorithms in Civil Engineering*, Springer

Yang, G.-P., Liu, S.-Y., Zhang, J.-K. and Feng, Q.-X. (2013), “Control and synchronization of chaotic systems by an improved biogeography-based optimization algorithm”, *Appl. Intel.*, **39**(1), 132-143.

Yang, X.-S. and Deb, S. (2009), “Cuckoo search via Lévy flights”, *Proceedings of World Congress on Nature & Biologically Inspired Computing*; *NaBIC*, Coimbatore, India, December.

Yetkin, M. (2015), *Düzlem çelik çerçevelerin sosyal örümcek optimizasyonuna göre boyutlandırılması* (*Optimum design of plane steel frames according to social spider optimization*), Balıkesir University, YÖK Tez Merkezi, Turkey.

Zhan, Z.-h., Chen, W.-n., Lin, Y., Gong, Y.-j., Li, Y.-l. and Zhang, J. (2013), “Parameter investigation in brain storm optimization”, *Proceedings of 2013 IEEE Symposium on Swarm Intelligence* (*SIS*), Singapore, April.

CC

Abbreviation list

3-D	:	Three Dimensional
ABC	:	Artificial Bee Colony
ACO	:	Ant Colony Optimization
BBO	:	Bio-Geography Based Optimization
BSO	:	Brain Storm Optimization
BtoC	:	Beam To Column
CtoC	:	Column To Column
CPSO	:	Conventional Particle Swarm Optimization
ELPSO	:	Enhanced Leader Particle Swarm Optimization
FFA	:	Firefly Algorithm
GA	:	Genetic Algorithm
GSA	:	Gravitational Search Algorithm
HSO	:	Harmony Search Optimization
LF	:	Levy Flight
LFABC	:	Artificial Bee Colony With LF Distribution
LFBBO	:	Bio-Geography Based Optimization With LF Distribution
LFBSO	:	Brain Storm Optimization With LF Distribution
LRFD-AISC	:	Load And Resistance Factor Design-American Institute Of Steel Construction
PSO	:	Particle Swarm Optimization

Symbol list

Bf_i^b	:	Flange width of the beam for the i^{th} beam-column connection
Bf_i^c	:	Flange width of the column for the i^{th} beam-column connection
$Center_{k,j}$	:	Center of k^{th} cluster for j^{th} design variable
C_i	:	Violation of the i^{th} constraint function
DL	:	Dead Load
D_i^c	:	Depth of a column for the i^{th} beam-column connection
D_i^b	:	Depth of lower-story column for the i^{th} column-column connection
D_i^a	:	Depth of upper-story column for the i^{th} column-column connection
F_i	:	Fitness value of the i^{th} design
FS	:	Number of food source in the ABC algorithm
g_i	:	i^{th} constraint function
HSI	:	High suitability index
$Iter$	:	Current iteration number in the optimization process
$Iter_{max}$	:	Maximum iteration number defined in the optimization method
l_i	:	Length of the i^{th} member
LFS	:	Limit Of Food Source
LL	:	Live Load
m_i	:	Mutation rate of i^{th} design for the BBO algorithm
m_{max}	:	Maximum mutation rate defined in the algorithm
M_{nx}	:	Nominal moment capacity for strong axis bending
M_{ny}	:	Nominal moment capacity for weak axis bending
M_{ux}	:	Ultimate moment occurred in the member for strong axis bending
M_{uy}	:	Ultimate moment occurred in the member for weak axis bending
n_{bc1}	:	Number of beam-column connections for connection type 1: beam is connected to the web of a column
n_{bc2}	:	Number of beam-column connections for connection type 2: beam is connected to the flange of a column
n_{bm}	:	Number of beam members
nc	:	Number of constraint function
n_{cc}	:	Number of column-column connections
NCl	:	Total Number of Clusters (For BSO Algorithm)
NED	:	Number of Elite Designs (For BBO Algorithm)

NG	:	Number of structural member groups (Number of design variables)
NH	:	Number of Habitat (For BBO Algorithm)
NI	:	Total number of Individuals (For BSO Algorithm)
NI_i	:	Number of individual in the *i*th cluster
n_j	:	Number of joints in the structure
N_{lc}	:	Number of load cases defined in the optimization problem
NM	:	Number of members in the structure
N_{sec}	:	Number of section defined in the optimization problem
n_{st}	:	Number of story in the structure
P_{S3}	:	Pre-determined probability used for changing centers of clusters (for BSO algorithm)
$P_{S4,1}$	:	Pre-determined probability used for selecting one cluster to generate new design (for BSO algorithm)
$P_{S4,2}$	:	Usage probability for the center of the selected cluster
$P_{Cl,i}$	:	Selection probability of cluster i (for BSO algorithm)
P_i	:	Selection probability of i^{th} structural design
P_{max}	:	Maximum selection probability in the optimization algorithm
P_n	:	Axial load capacity of the member
P_u	:	Ultimate axial force occurred in the member
rnd	:	Uniformly distributed random number between [0, 1]
tf_i^c	:	Flange thickness of the column for the i^{th} beam-column connection
SIV	:	Suitability index variable
SL	:	Step length values used for generating new individual (for BSO algorithm)
V	:	Total violation of the structural design
W	:	Weight of the structure
w_i	:	Weight per meter of section assigned to the member i
w_i^b	:	Unit weight of the lower column for the i^{th} column-column connection
w_i^a	:	Unit weight of the upper column for the i^{th} column-column connection
WL	:	Wind Load
$WL+$	:	Wind Load with positive Direction
$WL-$	:	Wind Load with negative Direction
WXL	:	Wind Load with global X Direction
WYL	:	Wind Load with global Y Direction
WL	:	Wind Load
W_p	:	Penalized weight of the structure
$\boldsymbol{x}$	:	Vector of the sequence number of sections assigned to the structure members
$x^{cand}{}_j$	:	Value of j^{th} design variable for candidate design
$X_{i,j}$	:	Value of j^{th} design variable for i^{th} design (individual)
α	:	Step size of the LF distribution which is relevant to the scales of the problem
β	:	Stability (Levy) index
ϕ_c	:	Factor of safety for compression
ϕ_b	:	Factor of safety for bending
δ	:	Computed deflection of the beam member
δ_{al}	:	Allowable deflection limit of the beam member
δi	:	Computed displacement of the i^{th} joint
δi_{al}	:	Allowable deflection limit
Δ^{is}	:	Computed inter-story drift
$\Delta^{is}{}_{al}$	:	Allowable inter-story drift
Δ^{top}	:	Computed top-story drift
$\Delta^{top}{}_{al}$	:	Allowable top-story drift
Γ	:	Gamma function
λ_i	:	Immigration rate of the i^{th} individual
μ_i	:	Emigration rate of the i^{th} individual
$\oplus$	:	Entry wise multiplications