10th World Congress on Structural and Multidisciplinary Optimization
May 19 -24, 2013, Orlando, Florida, USA

Optimum Design of Steel Space Frames via Bat Inspired Algorithm

Serdar Carbas[1], Oguzhan Hasancebi[2]

[1] Middle East Technical University, Department of Engineering Sciences, Ankara, Turkey, carbas@metu.edu.tr
[2]Middle East Technical University, Department of Civil Engineering, Ankara, Turkey, oguzhan@metu.edu.tr

1. Abstract
Design optimization of steel space frames is a very popular topic in structural engineering due to economy saved in cost of the structures by optimization process. Although the final cost of a steel frame is affected by many factors, such as material, manufacturing, erection and transportation costs, the material cost of steel comprises a great deal of overall cost of the structure. Hence, the design optimization of steel frames is focused on weight minimization in the literature based on the assumption that the use of least material leads to an economical design as well in terms of final cost of a structure. This study focuses on design optimization of steel space frames that are sized for minimum weight subject to stress, stability and nodal displacement and drift constraints according to Allowable Stress Design-American Institute of Steel Construction (ASD-AISC) specification. Bat inspired optimization (BIO) algorithm, which is a recently developed metaheuristic technique that exploits echolocation behavior of bats in searching a design space, is employed to deal with the optimization problem at hand. It is shown that BIO produces improved results with respect to other methods of metaheuristics.
2.Keywords: optimum design; discrete optimization; stochastic techniques; bat-inspired algorithm; steel frames.

3. Introduction
Designing a steel frame is among the usual tasks of a structural engineer. Both safety and economy have to be observed while designing a steel frame. The common practice is to observe structural safety always, while an economical design is pursued by the designer sometimes using intuition or experience, and occasionally using a trial-and-error process. However, despite the best effort of the designer, the optimum design cannot be reached in most cases, and even the design produced might sometimes be very far from economical range. The continuous inflation of the cost of materials necessitates the development of computer-aided numerical algorithms that are capable of optimizing the design of structures.
Optimization has been long studied through the globe in many disciplines of science and engineering. In the past structural optimization was overwhelmed with optimality criteria and mathematical programming based methods. Despite strong mathematical backgrounds and remarkable speed of convergence to the optimum, these methods have found limited applications in some optimization areas, such as discrete structural optimization The need for selection of member sizes from a list of ready sections hampers a direct application of these methods to practical structural optimization problems. Fortunately, in the last two decades a number of computational tools called metaheuristics have been developed, which makes it possible to find optimum solutions to problems from engineering practice [1]. Such tools are finding increasing industrial use due to their efficiency as well as ease in their implementations. These methods are recognized as one of the most practical approaches for solving many complex problems, and this is particularly true for many real-world problems that are combinatorial in nature. The practical advantage of metaheuristics lies in both their effectiveness and wide range of applicability [2]. The applications of these techniques to the structural optimization problems are very popular nowadays [3-6].
One of the most recent metaheuristic techniques is bat inspired optimization algorithm (BIO) developed by Yang [7]. The main idea behind BIO algorithm is to imitate echolocation behavior of bats. This algorithm is inspired from spectacular echolocation talent of bats, as these animals can find their prey and discriminate different types of insects even in complete darkness. The recent studies indicate that BIO algorithm produces superior results when solving engineering optimization problems [8-10]. In this paper, the BIO algorithm is formulated to solve minimum weight design problem of steel space frames subject strength and displacement provisions of ASD-AISC specification. The efficiency of the BIO algorithm is investigated and verified using a 3D industrial factory building under specified loadings.

4. Optimization of Steel Frames According to ASD-AISC
For a steel structure consisting of N_m members that are collected in N_d design groups (variables), the optimum design problem according to ASD-AISC [11] code yields the following discrete programming problem, if the design groups are selected from steel sections in a given profile list.
Find a vector of integer values $\mathbf{I}$ (Eq. 1) representing the sequence numbers of steel sections assigned to N_d member groups

$$\mathbf{I}^T = \left[I_1, I_2, \ldots, I_{N_d} \right] \tag{1}$$

to minimize the weight (W) of the frame

$$W = \sum_{i=1}^{N_d} \rho_i A_i \sum_{j=1}^{N_t} L_j \tag{2}$$

where A_i and ρ_i are the length and unit weight of the steel section adopted for member group i, respectively, N_i is the total number of members in group i, and L_j is the length of the member j which belongs to group i.
The members subjected to a combination of axial compression and flexural stress must be sized to meet the following stress constraints:

$$if \quad \frac{f_a}{F_a} > 0.15 \quad ; \quad \left[\frac{f_a}{F_a} + \frac{C_{mx} f_{bx}}{\left(1 - \frac{f_a}{F'_{ex}}\right) F_{bx}} + \frac{C_{my} f_{by}}{\left(1 - \frac{f_a}{F'_{ey}}\right) F_{by}} \right] - 1.0 \le 0 \tag{3}$$

$$\left[\frac{f_a}{0.60 F_y} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \right] - 1.0 \le 0 \tag{4}$$

$$if \quad \frac{f_a}{F_a} \le 0.15 \quad ; \quad \left[\frac{f_a}{F_a} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \right] - 1.0 \le 0 \tag{5}$$

If the flexural member is under tension, then the following formula is used instead:

$$\left[\frac{f_a}{0.60 F_y} + \frac{f_{bx}}{F_{bx}} + \frac{f_{by}}{F_{by}} \right] - 1.0 \le 0 \tag{6}$$

In Eqs. (3-6), F_y is the material yield stress, and $f_a=(P/A)$ represents the computed axial stress, where A is the cross-sectional area of the member. The computed flexural stresses due to bending of the member about its major (x) and minor (y) principal axes are denoted by f_{bx} and f_{by}, respectively. F_{ex}' and F_{ey}' denote the Euler stresses about principal axes of the member that are divided by a factory of safety of 23/12. F_a stands for the allowable axial stress under axial compression force alone, and is calculated depending on elastic or inelastic bucking failure mode of the member using Formulas 1.5-1 and 1.5-2 given in ASD-AISC [11]. The allowable bending compressive stresses about major and minor axes are designated by F_{bx} and F_{by}, which are computed using the Formulas 1.5-6a or 1.5-6b and 1.5-7 given in ASD-AISC [11]. It is important to note that while calculating allowable bending stresses, a newer formulation (Eq. (7)) of moment gradient coefficient C_b given in ANSI/AISC 360-05 [12] is employed in the study to account for the effect of moment gradient on lateral torsional buckling resistance of the elements,

$$C_b = \frac{12.5 M_{\max}}{2.5 M_{\max} + 3 M_A + 4 M_B + 3 M_C} R_m \le 3.0 \tag{7}$$

where M_{max}, M_A, M_B and M_C are the absolute values of maximum, quarter-point, midpoint, and three-quarter point moments along the unbraced length of the member, respectively, and R_m is a coefficient which is equal to 1.0 for doubly symmetric sections. C_{mx} and C_{my} are the reduction factors, introduced to counterbalance overestimation of the effect of secondary moments by the amplification factors ($1-f_a/F_e'$). For unbraced frame members, they are taken as 0.85. For braced frame members without transverse loading between their ends, they are calculated from $C_m=0.6-0.4(M_1/M_2)$, where M_1/M_2 is the ratio of smaller end moment to the larger end moment. Finally, for braced frame members having transverse loading between their ends, they are determined from the formula $C_m=1+\psi\ (f_a\ /\ F_e')$ based on a rational approximate analysis outlined in ASD-AISC [11] Commentary-H1, where ψ is a parameter that considers maximum deflection and maximum moment in the member.

For computation of allowable compression and Euler stresses, the effective length factors K are required. For beam and bracing members, K is taken equal to unity. For column members, alignment charts are furnished in ASD-AISC [11] for calculation of K values for both braced and unbraced cases. In this study, however, the following approximate effective length formulas are used based on Dumonteil [13], which are accurate to within about -1.0 and +2.0 % of exact results [14]:

For unbraced members:

$$K = \sqrt{\frac{1.6G_A G_B + 4(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \tag{8}$$

For braced members:

$$K = \frac{3G_A G_B + 1.4(G_A + G_B) + 0.64}{3G_A G_B + 2.0(G_A + G_B) + 1.28} \tag{9}$$

where G_A and G_B refer to stiffness ratio or relative stiffness of a column at its two ends.
It is also required that computed shear stresses (f_a) in members are smaller than allowable shear stresses (F_v), as formulated in Eq. (10).

$$f_v \leq F_v = 0.40 C_v F_y \tag{10}$$

In Eq. (10), C_v is referred to as web shear coefficient. It is taken equal to C_v =1.0 for rolled I-shaped members with $h/t_w \leq 2.24 E/F_y$, where h is the clear distance between flanges, E is the elasticity modulus and t_w is the thickness of web. For all other symmetric shapes, C_v is calculated from Formulas G2-3, G2-4 and G2-5 in ANSI/AISC 360-05 [12].
Apart from stress constraints, slenderness limitations are also imposed on all members such that maximum slenderness ratio ($\lambda = KL/r$) is limited to 300 for members under tension, and to 200 for members under compression loads. The displacement constraints are imposed such that the maximum lateral displacements are restricted to be less than H/400, and upper limit of story drift is set to be h/400, where H is the total height of the frame building and h is the height of a story.
Finally, we consider geometric constraints between beams and columns framing into each other at a common joint for practicality of an optimum solution generated. For the two beams B1 and B2 and the column shown in Figure 1, one can write the following geometric constraints:

$$\frac{b_{fb}}{b_{fc}} - 1.0 \leq 0 \quad \text{and} \quad \frac{b'_{fb}}{(d_c - 2t_f)} - 1.0 \leq 0 \tag{11) - (12}$$

where b_{fb}, b'_{fb}, and b_{fc} are the flange width of the beam B1, the beam B2 and the column, respectively, d_c is the depth of the column, and t_f is the flange width of the column. Eq. (11) simply ensures that the flange width of the beam B1 remains smaller than that of the column. On the other hand, Eq. (12) enables that flange width of the beam B2 remains smaller than clear distance between the flanges of the column (d_c-2 t_f).

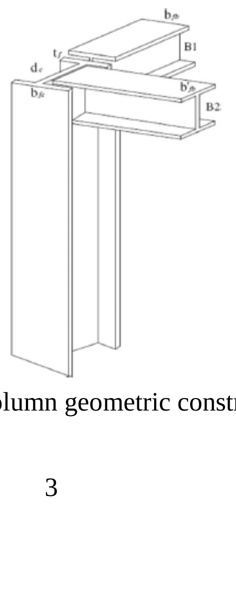


Figure 1: Beam-column geometric constraints.

5. Bat Inspired Optimization Algorithm

The bat-inspired optimization algorithm is derived from the echolocation behavior of bats. Echolocation is an advanced hearing based navigation system used by bats and some other animals to detect objects in their surroundings by emitting a sound to the environment. While they are hunting for preys or navigating, these animals produce a sound wave that travels across the canyon and eventually hits an object or a surface and return to them as an echo. The sound waves travel at a constant speed in zones where atmospheric air pressure is identical. By following the time delay of the returning sound, these animals can determine the precise distance to circumjacent objects. Further, the relative amplitudes of the sound waves received at each individual ear are used to identify shape and direction of the objects. The information collected this way of hearing is synthesized and processed in the brain to depict a mental image of their surroundings.

Yang [7] simulated echolocation behavior of bats and its associated parameters in a numerical optimization algorithm. However, in Hasançebi et al. [15] a major adaptation of the technique is carried out in its formulation and outline to generate an algorithm that performs efficiently for structural optimization problems. The basic steps in implementation of this BIO algorithm are described as follows.

Step 1. Initializing bat population (positions and velocities): A population of μ micro-bats (solutions) is randomly generated first, where μ refers to the population size. Each micro-bat $\mathbf{B}_i$ has two sets of components; a position (design) vector $\mathbf{x}_i$ and a velocity vector $\mathbf{v}_i$, Eq. (13).

$$\mathbf{B}_i=(\mathbf{x}_i, \mathbf{v}_i) \tag{13}$$

Step 2. Echolocation parameters and their initializations: Each micro-bat incorporates a set of echolocation parameters $\mathbf{\Omega}_i=(f_i,r_i,l_i)$, which consists of a frequency f_i, a pulse rate r_i, and a loudness parameter l_i. All the three echolocation parameters are non-negative dynamic real quantities with the following value ranges:

$$f_{min}\le f_i\le f_{max}\ ,\ r_{min}\le r_i\le r_{max}\ ,\ l_{min}\le l_i\le l_{max} \tag{14}$$

where f_{min} and f_{max} are the specified lower and upper bounds for the frequency parameter f_i, respectively; r_{min} and r_{max} are the specified lower and upper bounds for the pulse rate parameter r_i, respectively; and l_{min} and l_{max} are the specified lower and upper bounds for the loudness parameter l_i, respectively. Yang [7] states that the choice of upper and lower bounds for the echolocation parameters might have a significant influence on the convergence characteristics of the algorithm. In the present study the bounds f_{min}, $f_{\max}$, l_{max} and r_{min} are set to the following constants; f_{min}=0.0, $f_{\max}$=1.0, l_{max}=1.0, and r_{min}=0.5,whereas l_{min} and r_{max} are calculated using the following equations.

$$l_{\min}=\frac{1}{\sqrt{n_{\sec}}}\ ,\ r_{\max}=1-\frac{1}{n_d}\le 1.0 \tag{15}$$

In Eq. (15) n_{sec} is the number of sections in the discrete set used for sizing the design variables, and n_d is the number of discrete design variables. The echolocation parameters are initialized such that the initial frequency f_i^0 is set to a value randomly chosen between f_{min} and f_{max}. Besides, the initial loudness l_i is set to its maximum value l_{max}=1.0 whereas the initial pulse rate r_i is set to its minimum value r_{min}=0.5 for every micro-bat in the population.

Step 3. Evaluating micro-bats in the initial population: The initial population is evaluated. The objective function values of the feasible micro-bats that satisfy all problem constraints are directly calculated from Eq. (2). However, infeasible micro-bats that violate some of the problem constraints are penalized using an external penalty function approach, and their objective function values are calculated according to Eq. (16).

$$\phi=W\left[1+p\left(\sum_k c_k\right)\right] \tag{16}$$

In Eq. (16), Ø is the constrained objective function value, c_k is the k-th problem constraint and p is the penalty coefficient used to tune the intensity of penalization as a whole. This parameter is set to an appropriate static value, such as $p=1$.

Step 4. Storing the current population: The current population is stored and iteration counter t is increased by 1

Step 5. Generating candidate micro-bats: μ number of new micro-bats is generated as candidate solutions for the design population. This is implemented using a procedure that employs two probabilistic generation schemes referred to as random flying and local search. Random flying provides a more explorative search, allowing the micro-bat to fly to a new and possibly remote position in the search space. On the other hand, a more exploitative search is intended in a local search scheme, where a micro-bat selected from the current population

is perturbed in the close vicinity of its current solution to browse neighboring points. A new micro-bat is generated by applying either one of these two schemes, which is determined probabilistically. A uniform random number u_i is sampled between 0 and 1 for each micro-bat $\mathbf{B}_i$ in the current population, and it is compared with the pulse rate r_i of the micro-bat. If $u_i \geq r_i$, a new micro-bat is generated by flying $\mathbf{B}_i$ randomly to a new position in the design space. Otherwise, a micro-bat ($\mathbf{B}_k$ $k \in [1,\mu]$) is selected from the current population at random and a local search is performed around this solution to generate a new micro-bat. Probabilistically speaking, the odds of generating a new micro-bat using random flying and local search in this procedure are 1-r_{ave} and r_{ave}, respectively, where r_{ave} represents the average pulse rate of the micro-bats in the current population.
It should be noted that the initial value of pulse rate r_i^0 is set to 0.5 in the algorithm and it increases towards a value around r_{max}=1-1/n_d in the course of the search process. It follows that in the beginning of the search the new candidate solutions are originated using the two generation schemes under equal probability. However, as the search goes on, the role of local search is augmented while that of random flying is diminished. This way an exploitative search is progressively dominated in time to benefit more from the previously visited good solutions than exploring new design regions of the search space.
Random Flying: A new candidate solution is generated from a micro-bat $\mathbf{B}_i$ through random flying by adjusting its frequency f_i^0 first and updating its velocity and position next. Unlike real bats which exhibit random motion patterns, a micro-bat follows certain rules for the velocity and position update, which are formulated in Eq. (17)

$$f_i^t = f_{min} + (f_{max} - f_{min})u_i \text{ , } \mathrm{v}_i^t = round\,[\mathrm{v}_i^{t-1} + (\mathrm{x}_i^{t-1} - \mathrm{x}_*)\, f_i^t] \text{ , and } \mathrm{x}_i^t = \mathrm{x}_i^{t-1} + \mathrm{v}_i^t \quad (17)$$

In Eq. (13), f_{min} and f_{max} are the lower and upper bounds imposed for the frequency range of micro-bats, respectively and $u_i \in [0,1]$ is a random number sampled anew for each micro-bat according to a uniform distribution. In Eq. (17), v_i^t and v_i^{t-1} are the velocity vectors of the *i*-th micro-bat at time steps (iterations) *t* and *t*-1, respectively; likewise x_i^t and x_i^{t-1} are the position vectors of the micro-bat at iterations *t* and *t*-1, respectively and x_* is the current global best solution representing the best-so-far solution found during the optimization process.
Local Search: A local search is implemented on a randomly selected micro-bat $\mathbf{B}_k$ from the current population. In the original bat-inspired algorithm developed by Yang [7] for continuous variable optimization problems, the local search is implemented using Eq. (18)

$$x_i^t = x_i^{t-1} + \xi_{i,j}\, l_{ave}^{\ t} \quad (18)$$

where $\xi_{i,j}$ is a uniform random number between -1 and 1 selected anew for each design variable *j* of the micro-bat $\mathbf{B}_k$. A reformulation of this equation is carried out for discrete structural optimization problem as formulated in Eqs. (19) and (20).

$$\xi_{k,j} = N(0,\sigma).\sqrt{n_{\mathrm{sec}}} \quad (19)$$

$$x^t_{\ k,j} = \begin{cases} x^{t-1}_{\ \ k,j} + round\ \ \xi_{k,j}.l_{ave}^{\ \ t-1} & \text{: if } u_{k,j} \geq r_k \\ x^{t-1}_{\ \ k,j} & \text{: if } u_{k,j} < r \end{cases} \quad (20)$$

In Eqs. (19) and (20), N(0,σ) is a normally distributed random with mean 0 and standard deviation σ; n_{sec} is the number of sections in the discrete set used for sizing the design variables; $\mathrm{x}_{k,j}^t$ and $\mathrm{x}_{k,j}^{t-1}$ are the values of *j*-th design variable in the micro-bat $\mathbf{B}_k$ at time steps *t* and *t*-1, respectively; and r_k is the pulse rate of the micro-bat $\mathbf{B}_k$. The numerical experiments performed on several test problems indicated that the standard deviation σ can be taken as 1.0 for problems with small to medium size design spaces, and 2.0 or higher for problems with larger size design spaces.
The rationale behind using a normal distribution in Eq. (19) is to facilitate occurrences of small step sizes as compared to large ones during local search. Besides, the term √(n_{sec}) in this equation is used to adjust the extent of the region scanned by the algorithm during local search in relation to the size of the discrete set. It can be noted that the algorithm permits larger step sizes, as the size of the discrete set increases.
It should also be noted that unlike Eq. (18) where all the design variables are subjected to transition (perturbation) during local search, Eq. (20) motivates transitions over a selected number of design variables, which is indeed controlled probabilistically by the pulse rate. Recalling that pulse rate is initially set to 0.5 for all micro-bats, a maximum of 50% of the design variables is then perturbed on average for each micro-bat at the start, and this ratio decreases to 1-r_k in connection with an increase in pulse rate as the search continues. This

way, while the algorithm is converging towards the optimum the number of design variable transitions is also restricted progressively towards a more exploitative local search achieved by reduced search dimension. This can be reasoned by the fact that unlike continuous variable optimization, structural optimization problems may be highly sensitive to the changes in design variables due to discrete nature of the sizing variables. That is to say, even small changes in a few design variables may yield a solution with entirely different structural behavior. Especially this becomes a more critical issue when the algorithm is converging towards the optimum since the optimum lies on or near the constraint boundaries in almost all practical applications of structural optimization. Design transitions over many design variables at these stages generally lead to large or uncontrolled step sizes in discrete design space, resulting in either infeasible or unsatisfactory design points. Hence, it is essential to limit the number of design variable transitions in order to generate successful moves when approaching towards the optimum.

Step 6. Evaluating candidate micro-bats: The objective function values of the feasible and infeasible candidate micro-bats are calculated from Eqs. (2) and (16), respectively.

Step 7. Echolocation Parameters Update: After evaluating candidate micro-bats, the echolocation parameters are updated for improving candidates that move to better points than before. The rationale behind this is to automatically adopt a more useful set of values for the echolocation parameters, similar to real bats which adjust those parameters based on the distance to the target object. In the original BIO algorithm developed by Yang [7], this is performed by comparing the micro-bat with the global best design, which refers to the solution with the minimum objective function value located so far by the entire micro-bat population. Accordingly, every time when the global best design is improved by a candidate micro-bat $\mathbf{B}_i$, a uniform random number u_i is sampled in the range [0,1] and if it is smaller than the pulse rate l_i of the micro-bat, then its echolocation parameters r_i, l_i are updated using the following equations:

$$l_i^{'} = \alpha \,.\, l_i \text{ and } r_i^{t+1} = r_{max}[1\text{-exp}(\gamma t)] \qquad (21)$$

where, $l_i^{'}$ and l_i are the previous and updated values of the loudness for micro-bat $\mathbf{B}_i$, t is iteration number, r_i^{t+1} is the pulse rate of the micro-bat $\mathbf{B}_i$ at iteration t+1, r_{max} is the maximum value of the pulse rate, and finally α and γ are the adaptation parameters of loudness and pulse rate, respectively.

In the BIO algorithm employed for structural optimization problems here, two modifications have been carried out regarding this update methodology. First of all, a micro-bat is allowed to update its echolocation parameters each time when it produces a solution that surpasses its individual best, not the global best necessarily. The individual best refers to the best solution attained by the micro-bat itself during its iteration history. Unlike improving the global best, the latter is much easily and frequently achieved by all micro-bats enabling a recurrent echolocation parameter update during the search. Secondly, a reformulation of Eq. (21) is carried out for adaptation of pulse rate parameter as given in Eq. (22).

$$r_i^{t+1} = 1\text{-}(1\text{-}\, r_i^{0})\gamma^{t+1} \leq r_{max} \qquad (22)$$

Eq. (22) facilitates a more gradual change of pulse rate parameter from its initial (minimum) value of r_i^0 towards r_{max}, whereas in Eq. (21) the pulse rate immediately approaches r_{max} in a few iterations and remains stationary at this value thereafter.

Step 8. Selection: Selection is then carried out between current and candidate micro-bats to form the members of the next population which will parent and guide the generation of subsequent micro-bats. The selection methodology employed in the BIO algorithm is borrowed from the well-known variant of evolution strategies technique referred to as (μ+μ)-ES [16]. In this selection methodology current and candidate micro-bats are set into competition together and the best μ solutions from a total of μ+μ=2μ current and candidate solutions are selected deterministically in reference to their objective function values.

Step 9. Termination: The steps 4 through 8 are implemented in the same way until a termination criterion is met.

6. Design Example

In this section the performance of the BIO algorithm is investigated using a 209-member industrial factory building (Figure 2) consisting of 100 joints. The structure is composed of 100 joints, and the 209 members of the frame are collected in 14 member groups (sizing design variables). The frame is sized for minimum weight by selecting the members from a set of discrete section presenting practical design application instance of real-size problems according to provisions of ASD-AISC [11] specification. This design problem has been studied formerly using other metaheuristic techniques, including classical harmony search (CHS) [17], adaptive harmony search (AHS) [17], big bang-big crunch (BB-BC) [18], exponential big bang-big crunch (EBB-BC) [18], and modified big bang-big crunch (MBB-BC) [18]. Therefore, comprehensive comparisons are provided between the optimum solutions obtained for this problem using the BIO algorithm and other metaheuristics.

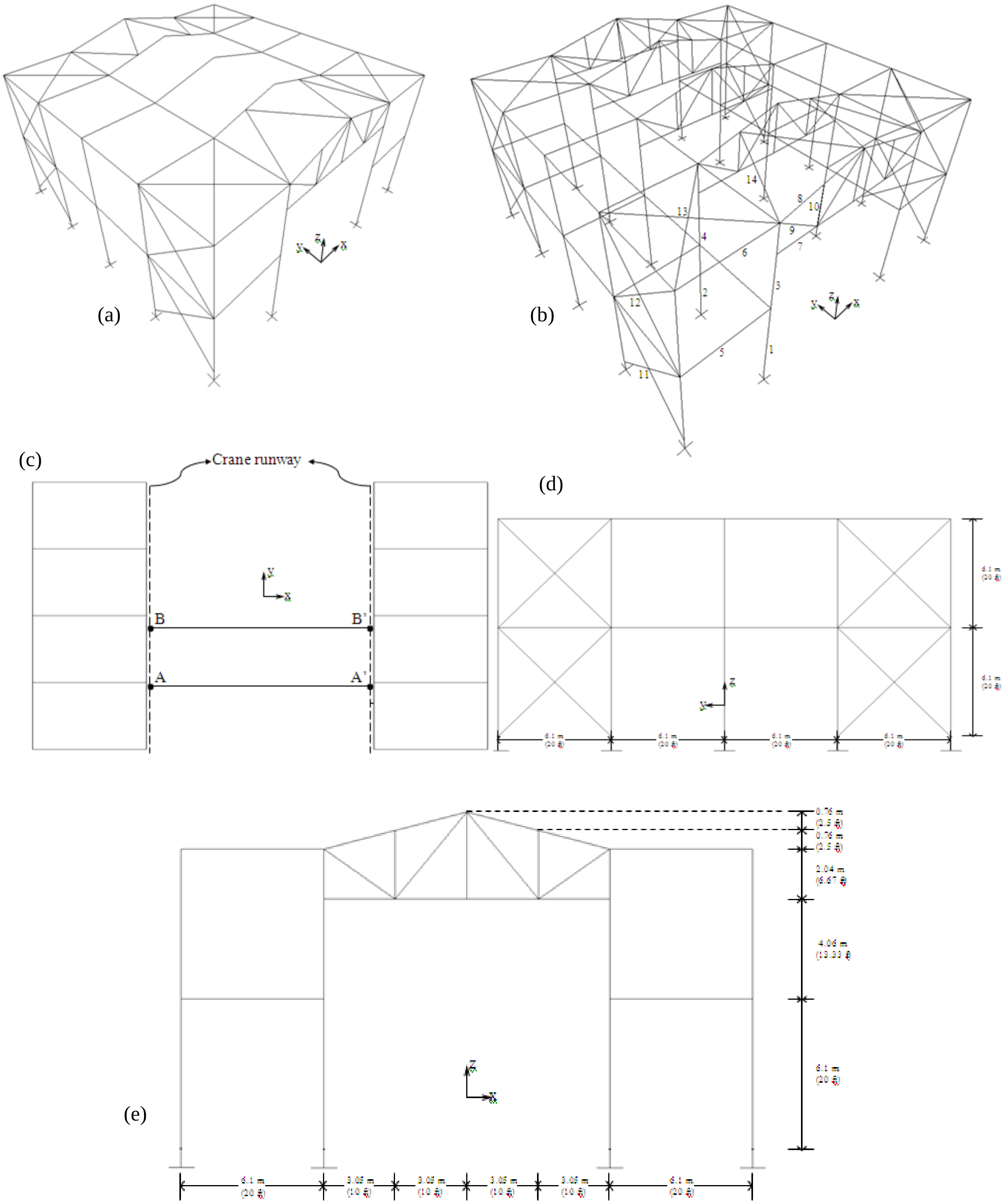
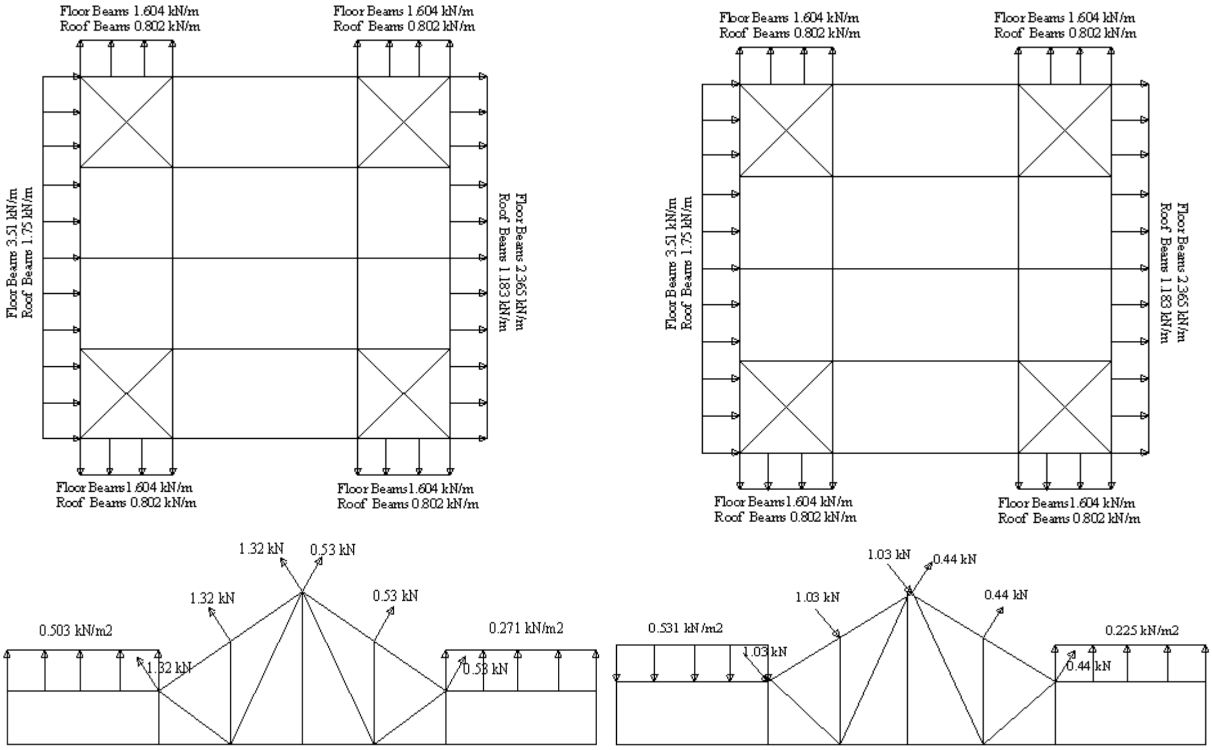


Figure 2: 209-member industrial factory building a) 3D view, b) Member grouping, c) First floor plan view, d) Side view, e) Front view

For a fair comparison of results, the maximum number of iterations is limited to 1000. During numerical implementations, the control parameters of the BIO algorithm are chosen as follows: population size μ=50, minimum frequency f_{min}=0.0, maximum frequency f_{max}=1.0, initial (maximum) loudness l_i^0=l_{max}=1.0, loudness adaptation parameter α=0.95, initial (minimum) pulse rate r_i^0=r_{min}=0.5, pulse rate adaptation parameter γ=0.98, standard deviation σ=2.0, and finally penalty coefficient *p*=1. The material properties of steel taken as follows: modulus of elasticity (*E*)=29,000ksi (203,893.6MPa) and yield stress (F_y)=36ksi(253.1MPa). Further, displacements of all the joints in x and y directions are limited to 3.43 cm (1.25 in.), and the maximum allowable value of inter-story drifts is taken as 1.52 cm (0.6 in.).



a) The WL1 for roof and floor beams

b) The WL2 for roof and floor beams

Figure 3: The two wind load cases considered for the design of 209-member industrial factory building.

For the design of this industrial building three different types of loads namely dead, crane and wind loads are considered. A design dead load of 1.2 kN/m2 (25.06 lb/ft2) is assumed to be acting on both floors of the side frames, resulting in uniformly distributed loads of 14.63 kN/m (1004.55 lb/ft) and 7.32 kN/m (502.27 lb/ft) on the interior and exterior beams of the side frames. Here, the dead weights of the gable roofs are neglected due to relatively light weight of these components. The crane load is modeled as two pairs of moving live loads acting on both sides of the crane runway beams. Each pair consists of a concentrated load of 280 kN (62.9 kip) and a couple moment of 75 kN m (5532 kip.ft). The crane load is represented in two distinct load cases as CL1 and CL2 by selecting two different positions for the crane on its runway. As shown in Figure 2(c), in CL1, the crane is positioned at points A and A’ to create maximum effect on the second framework. However, in CL2, it is positioned in the middle of the runway beam between the second and third frameworks (shown as B and B’ in Figure 2(c)) to maximize the response in the beams directed along y-axis. For design purpose, only the wind in the x-direction is considered and the corresponding wind forces are computed based on a basic wind speed of V = 46.94 m/s (105 mph) in line with the prescriptions given in ASCE 7-05 [19]. As shown in Figure 3, two load cases referred to as WL1 and WL2 are generated based on the sign of the internal wind pressure exerted on the external faces of the building. In both cases, it is assumed that wind produces a positive compression pressure on windward face, while it causes a negative suction effect on leeward face as well as on side walls of the building. In WL1 the suction effect is considered for the entire roof surface, while in WL2 one part of the roof is subjected to compression pressure. Amongst the five load cases (DL, CL1, CL2, WL1 and WL2), a total of six load combinations are generated for the strength design of structural members according to ASD-AISC [11] provisions, as follows: (i) 1.0DL + 1.0CL1, (ii) 1.0DL + 1.0CL1 + 1.0WL1, (iii) 1.0DL + 1.0CL1 + 1.0WL2, (iv) 1.0DL + 1.0CL2, (v) 1.0DL + 1.0CL2 + 1.0WL1, (vi) 1.0DL + 1.0CL2 + 1.0WL2.

Table 1: Comparison of results for 209-member industrial factory building.

Sizing variables	HS	AHS	BB-BC	EBB-BC	MBB-BC	BIO
1	W8X31	W8X31	W16X57	W10X33	W10X33	W10X33
2	W12X40	W10X39	W16X57	W10X33	W10X33	W10X33
3	W8X31	W12X26	W8X28	W8X24	W12X26	W8X24
4	W8X40	W8X40	W21X68	W10X33	W8X31	W10X33
5	W24X62	W24X62	W24X62	W24X62	W24X62	W24X62
6	W12X26	W10X26	W21X44	W12X26	W12X26	W12X26
7	2L2.5X2X3/16	2L2X2X1/8	2L5X5X5/8	2L2X2X1/8	2L2X2X1/8	2L2X2X1/8
8	2L2X2X1/8	2L2X2X1/8	2L2X2X1/8	2L2X2X1/8	2L2.5X2.5X3/16	2L2X2X1/8
9	2L3X3X3/16	2L3X3X3/16	2L4X4X5/8	2L3X3X3/16	2L4X3.5X1/4	2L3X3X3/16
10	2L3X2.5X5/16	2L2X2X1/8	2L2.5X2.5X3/16	2L2X2X1/8	2L2X2X3/16	2L2X2X1/8
11	2L6X6X7/16	2L6X6X5/16	2L6X6X3/4	2L6X6X5/16	2L6X6X3/8	2L6X6X5/16
12	2L6X6X3/8	2L6X6X5/16	2L8X8X3/4	2L6X6X5/16	2L6X6X3/8	2L6X6X5/16
13	2L6X6X5/16	2L6X6X5/16	2L6X6X5/8	2L6X6X5/16	2L6X6X7/16	2L6X6X5/16
14	2L6X6X5/16	2L5X5X5/16	2L5X5X7/16	2L5X5X5/16	2L5X5X3/8	2L5X5X5/16
Weight, lb (kg)	102924.73 (46685.83)	97121.3 (44053.45)	161764.99 (73375.37)	94631.38 (42924.07)	101842.77 (46195.10)	94631.38 (42924.07)

The BIO algorithm is employed to minimize the weight of the industrial factory building. In Table 1 the minimum weight designs of the structure obtained by this algorithm is compared to the previously reported results [17, 18] using harmony search (HS) and its adaptive variant (AHS) techniques, big bang-big crunch and its modified and exponential versions (BB-BC, MBB-BC and EBB-BC). The BIO algorithm performs very well and produces the best known solution of the problem, which is 42924.07 kg (94631.38 lb). The very same design has been formerly attained for the problem by EBB-BC algorithm. The optimum designs attained with AHS and HS techniques happened to be 44053.45 kg (97121.3 lb) and 46685.83 kg (102924.73 lb), respectively. The optimum design weight acquired with MBB-BC technique was 46195.10 kg (101842.77 lb). A substandard performance was exhibited by BB-BC algorithm, in which the structural weight could only be decreased to 73375.37 kg (161764.99 lb) due to stagnation of the algorithm in a local optimum relatively in the early stage of the search process. Figure 4 shows the variation of the best feasible design obtained so far in the search processes using different metaheuristics.

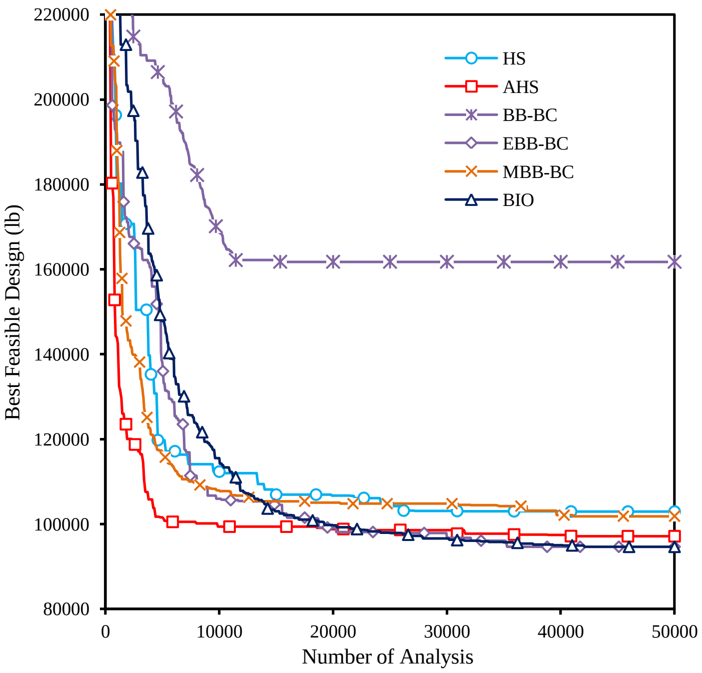


Figure 4: Design history graph of 209-member industrial factory building.

7. Conclusions

In this study a bat-inspired optimization algorithm, BIO, has been introduced as an effective method for optimization of steel space frames with discrete sizing variables. The algorithm employs basic principles of bat inspired technique, yet a thorough reformulation of the technique is carried out for its application to structural optimization. The efficiency of the resulting algorithm is numerically examined using a 209-member industrial building. In this test problem the performance of the BIO algorithm is measured against a variety of different metaheuristic techniques under the same design considerations. A comparison of numerical results attained using different metaheuristics clearly proves the efficiency of the BIO algorithm in structural optimization. Apparently, the robustness of the BIO algorithm lies in its enhanced ability in achieving a satisfactory tradeoff between two contradictory requirements of the search process known as exploration and exploitation, which are characterized by the algorithm as random flying and local search. The algorithm achieves this by implementing the echolocation parameters of pulse rate and loudness in an efficient manner during the search process. In the beginning of the optimization process the roles of explorative and exploitative search are balanced in an identical weight by setting pulse rate parameter to 0.5. As the iterations go on, the role of exploitative search becomes more prominent in proportion to increase in pulse rate parameter, while the loudness is decreased in the meantime to gradually narrow the size of the area investigated during local search.

8. References

[1] P.W. Christensen and A. Klarbing, *An introduction to structural optimization*, Springer, 2009.

[2] S. Olafsson, *Metaheuristics*, in Nelson and Henderson (Eds.), Handbook on Simulation, Handbooks in Operations Research and Management Science VII, Elsevier, 633-654, 2006.

[3] L. Lamberti and C. Pappalettere, *Metaheuristic design optimization of skeletal structures: a review*, in B.H.V. Topping (Ed.), Computational Technology Reviews, 4, 1-32, 2011.

[4] M.P. Saka, *Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review*, in B.H.V. Topping (Ed.), Civil Engineering Computations: Tools and Techniques, Saxe-Coburg Publications, Stirlingshire, UK, 105-147, 2007.

[5] M.P. Saka and E. Dogan, *Recent developments in metaheuristic algorithms: a review*, in B.H.V. Topping (Ed.), Computational Technology Reviews, 5, 31-78, 2012.

[6] M.P. Saka, E. Dogan, and I. Aydogdu, *Review and Analysis of Swarm-Intelligence Based Algorithms*, in Yang, Cui, Xiao and Gandomi (Eds.), Swarm Intelligence and Bio-Inspired Computation: Theory and Applications, Elsevier, 2013.

[7] X-S Yang, *A new metaheuristic bat-inspired algorithm*. in J.R. Gonzalez et al. (Eds), Nature inspired cooperative strategies for optimization (NISCO 2010), Studies in computational intelligence, 284, 65-74, Springer, Berlin, 2010.

[8] A.H. Gandomi, X-S Yang, A.H. Alav, and S. Talatahari, Bat algorithm for constrained optimization tasks, *Neural Computing and Applications*, DOI: 10.1007/s00521-012-1028-9, 2012.

[9] X-S Yang, Bat Algorithm for Multiobjective Optimization, *International Journal of Bio-Inspired Computation*, 3, 5, 267-274, 2011.

[10] X-S Yang and A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, *Engineering Computations*, 29, 5, 464 – 483, 2012.

[11] ASD-AISC, *Manual of Steel Construction-Allowable Stress Design*, 9th edition, Chicago, Illinois, USA, 1989.

[12] ANSI/AISC 360-05, *Specification for Structural Steel Buildings*, Chicago, Illinois, USA, 2005.

[13] P. Dumonteil, Simple Equations for Effective Length Factors, *Engineering Journal*, AISC, 29, 3, 111-115, 1992.

[14] J. Hellesland, *Review and Evaluation of Effective Length Formulas*, Research Report, 94-2, University of Oslo, 1994.

[15] O. Hasancebi, T. Teke, and O. Pekcan, A bat-inspired algorithm for structural optimization, *Computers and Structures*, under evaluation, 2013.

[16] T. Bäck, *Evolutionary algorithms in theory and practice*, New York Oxford University Press, 1996.

[17] M.P. Saka and O. Hasancebi, *Adaptive harmony search algorithm for design code optimization of steel structures*. in Z. Geem (Ed.), Harmony search algorithms for structural design optimization, Berlin, Heidelberg: Springer-Verlag, 79–120, 2009.

[18] O. Hasancebi and S. Kazemzadeh Azad, An exponential big bang-big crunch algorithm for discrete design optimization of steel frames, *Computers and Structures*, 110–111, 167–179, 2012.

[19] ASCE 7–05, *Minimum design loads for buildings and other structures*, ASCE, 2005.