Computers and Structures 128 (2013) 77–90

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

A bat-inspired algorithm for structural optimization

O. Hasançebi *, T. Teke, O. Pekcan

Middle East Technical University, Dept. of Civil Eng., Ankara, Turkey

ARTICLE INFO

Article history:
Received 13 December 2012
Accepted 13 July 2013

Keywords:
Structural optimization
Discrete optimization
Metaheuristic search techniques
Bat-inspired search
Steel trusses

ABSTRACT

Bat-inspired (BI) search is a recently developed numerical optimization technique that makes use of echolocation behavior of bats in seeking a design space. This study intends to explore capabilities and potentials of this newly developed method in the realm of structural optimization. A novel algorithm is developed that employs basic principles of this method for structural optimization problems specifically. Performance of the proposed algorithm is measured using one benchmark as well as three practical truss structures that are sized for minimum weight subject to stress, stability and displacement constraints according to American Institute of Steel Construction-Allowable Stress Design (AISC-ASD) specification. The numerical results demonstrate efficiency of the proposed algorithm in practical structural optimization.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The ongoing research in optimization has led to the development of new optimization approaches that exhibit certain advantages over more traditional techniques in various aspects. The need for developing new optimization approaches stemmed from the fact that the traditional techniques such as mathematical programming approaches, which have been used overwhelmingly for solving many engineering optimization problems in the past, have remained either inefficient or insufficient when solving today's engineering problems of real-life. The large scale of engineering systems, presence of numerous design variables and practical requirements of designs in actual applications complicated the present day problems to the point that makes it difficult to handle using traditional optimization methods. In general, traditional optimization methods suffer from employing a gradient based search algorithm that necessitates a continuous design space and seeks local optima rather than the global optimum.

Recently, a group of optimization techniques referred to as metaheuristic search algorithms have emerged to be robust tools for dealing with today's engineering problems of increased complexity. The rising popularity of these techniques derives from (i) the lack of dependency on gradient information; (ii) inherent capability to deal with both discrete and continuous design variables; and (iii) automated global search features to produce near-optimum solutions (if not the global optimum) for complicated problems. These novel and innovative approaches are derivative-free methods and make use of the ideas inspired from the nature. The basic idea behind these techniques is to simulate biological and physical systems in nature, such as natural evolution, immune system, swarm intelligence, annealing process, etc., in a numerical algorithm [1–6]. The key difference between these algorithms lies in the way the moves in the design space are proposed based on an associated nature-inspired strategy. There are many metaheuristic techniques available in the literature nowadays. A detailed review of these algorithms as well as their applications in engineering optimization problems can be found in Lamberti and Pappalettere [7], Saka [8], and Saka and Doğan [9].

One latest addition to metaheuristic algorithms is the bat-inspired (BI) search, which was recently proposed by Yang [10]. The bat-inspired search makes use of echolocation behavior of bats in searching for the optimum. In simple words, echolocation is used to refer to the way bats use to navigate their surroundings. Bats get to find their directions and detect prey and different types of objects around them even in complete darkness. They achieve this by emitting calls out to the environment and listening to the echoes that bounce back from them. They can identify location of other objects and instinctively measure how far they are away from them by following delay of the returning sound. Yang [10] idealized such echolocation behavior of bats and turned it into a numerical algorithm for optimization problems. Efficiency of the resulting algorithm was validated and compared with other existing algorithms using some single and multi-objective standard functions of unconstrained optimization in Yang [10,11], respectively. Besides, performance of the technique in benchmark problems of constrained engineering optimization was investigated in Yang and Gandomi [12] and Gandomi et al. [13]. The results obtained in these studies have clearly documented superiority of the bat-inspired search over other techniques.

* Corresponding author. Tel.: +90 312 2102456; fax: +90 312 2105401.
E-mail address: oguzhan@metu.edu.tr (O. Hasançebi).

0045-7949/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2013.07.006

This paper investigates the use and application of bat-inspired search technique in the realm of structural optimization. The objective of the study is to unveil capabilities and potentials of this newly developed method in practical problems of structural optimization. A novel bat-inspired (BI) algorithm is developed that employs fundamental principles of the technique to solve structural optimization problems with discrete sizing variables. The performance of the proposed algorithm is experimented and evaluated using one benchmark and three practical truss structures that are sized for minimum weight subject to stress, stability and displacement constraints according to American Institute of Steel Construction-Allowable Stress Design (AISC-ASD) specifications. Optimized designs obtained with the proposed BI are compared with those of other metaheuristic techniques. The results obtained verify promising performance of the proposed algorithm in structural optimization problems.

The remaining sections of the paper are organized as follows. The second section covers a mathematical statement of the structural optimization problem considered. In the third section the proposed bat-inspired algorithm is introduced with particular emphasis on its reformulations for discrete structural optimization problems. Numerical examples proving efficiency of the proposed algorithm are covered in the fourth section. A brief conclusion of the study is given in the last section.

2. Optimum size design problem of steel trusses

Typically in practical design optimization of truss structures the aim is to find a minimum cost or weight design by selecting cross-sectional areas of structural members from a table of available sections such that the final design satisfies strength and serviceability requirements determined by technical standards. For a given truss structure composed of N_m members grouped into N_d sizing design variables, the optimization problem can be formulated as follows.

2.1. Objective function

The objective is to find a vector of integer values $\mathbf{I}$ (Eq. (1)) representing the sequence numbers of standard sections in a given section table,

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

to generate a vector of cross-sectional areas $\mathbf{A}$ (Eq. (2)) for N_m members of the structure,

$$\mathbf{A}^T = [A_1, A_2, \ldots, A_{N_m}] \tag{2}$$

such that $\mathbf{A}$ minimizes the following weight objective function:

$$W = \sum_{m=1}^{N_m} \rho_m L_m A_m \tag{3}$$

where W is the weight of the structure, ρ_m, L_m, A_m are unit weight, length, and cross-sectional area of the mth member, respectively.

2.2. Design constraints

The design constraints consist of the following limitations imposed on overall structural response and behavior of individual members:

$$g_m = \frac{\sigma_m}{(\sigma_m)_{all}} - 1 \leqslant 0; \quad m = 1, \ldots, N_m \tag{4}$$

$$s_m = \frac{\lambda_m}{(\lambda_m)_{all}} - 1 \leqslant 0; \quad m = 1, \ldots, N_m \tag{5}$$

$$\delta_{jk} = \frac{d_{j,k}}{(d_{j,k})_{all}} - 1 \leqslant 0; \quad j = 1, \ldots, N_j \tag{6}$$

In Eqs. (4)–(6), functions g_m, s_m and δ_{jk} are the optimization constraints on stresses, slenderness ratios, and displacements, respectively; σ_m and $(\sigma_m)_{all}$ are the computed and allowable axial stresses for mth member, respectively; λ_m and $(\lambda_m)_{all}$ are the slenderness ratio and its upper limit for mth member, respectively; N_j is the total number of joints; and finally $d_{j,k}$ and $(d_{j,k})_{all}$ are the displacements computed in kth direction of jth joint and its allowable value, respectively. In the present study, these constraints are implemented according to American Institute of Steel Construction-Allowable Stress Design (AISC-ASD) [14] code specifications. The complete details of design formulations can be found in Hasançebi et al. [15].

3. Bat-inspired algorithm for structural optimization

The bat-inspired 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 returns to them as an echo. The sound waves travel at a constant speed in zones where atmospheric air pressure is identical. By following time delay of the returning sound, these animals can determine the precise distance to circumjacent objects. Further, relative amplitudes of the sound waves received at each individual ear are used to identify shape and direction of the objects. The information collected with this way of hearing is synthesized and processed in the brain to depict a mental image of their surroundings.

In general echolocation calls are characterized by three features; namely pulse frequency, pulse emission rate and loudness (intensity). When flying, bats emit echolocation calls with varying frequencies between 25 kHz and 150 kHz depending on proximity of the target [10]. Although low frequency sounds travel further than high-frequency sounds, the latter give bats more detailed and distinctive information about the surrounding objects. On the other hand, the pulse rate corresponds to the number of pulses emitted per second and it can also be adjusted by bats according to the distance from the target object. It is known that bats increase the rate of pulse to 200 pulses per second when approaching a target. Finally, bats decrease the intensity (loudness) of pulse from 120 dB (loudest) to 50 dB (quietest) as they come closer to their prey. Yang [10] simulated echolocation behavior of bats and its associated parameters in a numerical optimization algorithm.

It is important to mention that metaheuristic search techniques offer a general solution methodology for a wide range of optimization problems from different disciplines. On the other hand, each optimization problem has unique features of its own, and in most cases a problem-wise reformulation is necessary to achieve the best performance of an algorithm for a particular class of problems. In the following a novel bat-inspired algorithm developed specifically for structural optimization problems with discrete sizing variables is introduced. This algorithm employs the fundamental principles of echolocation behavior of bats and thus it has strong similarities with the numerical optimization algorithm developed by Yang [10] in this sense. However, 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 the proposed BI 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. (7). The position vector retains values (positions) of the design variables, while the velocity vector is used to vary these positions during the search. The iteration counter t is set to zero initially:

$$\mathbf{B}_i = (\mathbf{x}_i, \mathbf{v}_i) \tag{7}$$

Step 2. *Echolocation parameters and their initializations*: The next step is to initialize an echolocation parameter set for the micro-bats in the population. 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:

$$f_{\min} \leqslant f_i \leqslant f_{\max}, \quad r_{\min} \leqslant r_i \leqslant r_{\max}, \quad l_{\min} \leqslant l_i \leqslant l_{\max} \tag{8}$$

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 [10] states that the choice of upper and lower bounds for echolocation parameters might have a significant influence on 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:

$$f_{\min} = 0.0, \quad f_{\max} = 1.0, \quad l_{\max} = 1.0, \quad \text{and} \quad r_{min} = 0.5 \tag{9}$$

whereas $l_{\min}$ and $r_{\max}$ are proposed to be calculated using the following equations:

$$l_{\min} = \frac{1}{\sqrt{n_{\text{sec}}}} \tag{10}$$

$$r_{\max} = 1 - \frac{1}{N_d} \leqslant 1.0 \tag{11}$$

In Eqs. (10) and (11), 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^0 is set to its maximum value $l_{\max} = 1.0$ whereas the initial pulse rate r_i^0 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, where all micro-bats are analyzed with the set of steel sections selected for the design variables, and force and deformation responses are obtained under the loads. Objective function values of the feasible micro-bats that satisfy all problem constraints are directly calculated from Eq. (3). 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. (12):

$$\phi = W\left[1 + p\left(\sum_k c_k\right)\right] \tag{12}$$

In Eq. (12), ϕ is the constrained objective function value, c_k is the kth 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, which consists of the best micro-bats (solutions) located so far in the search process from the beginning, is stored and iteration counter t is increased by one.

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 a 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 using the following pseudo code.

i: = 1
repeat
  if ($u_i \geqslant r_i$) **then**
    – Select ith micro-bat $\mathbf{B}_i$ from the current population
    – Fly $\mathbf{B}_i$ to a new position randomly
  else
    – Select any micro-bat $\mathbf{B}_k$ from the current population, $k \in [1, \mu]$
    – Perform local search around $\mathbf{B}_k$
  i: = i + 1
until ($i > \mu$)

In the pseudo code shown above, 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 $u_i \geqslant r_i$, 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 proposed 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 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. In this way, exploitative search progressively dominates 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 Eqs. (13)–(15):

$$f_i^t = f_{\min} + (f_{\max} - f_{\min})u_i \tag{13}$$

$$\mathbf{v}_i^t = round\left[\mathbf{v}_i^{t-1} + \left(\mathbf{x}_i^{t-1} - \mathbf{x}_*\right)f_i^t\right] \tag{14}$$

$$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} + \mathbf{v}_i^t \tag{15}$$

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 Eqs. (14) and (15), $\mathbf{v}_i^t$ and $\mathbf{v}_i^{t-1}$ are the velocity vectors of the ith micro-bat at time steps (iterations) t and $t-1$, respectively; likewise $\mathbf{x}_i^t$ and $\mathbf{x}_i^{t-1}$ are the position vectors of the micro-bat at iterations t and $t-1$, respectively and $\mathbf{x}_*$ 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 [10] for continuous variable optimization problems, the local search is implemented using Eq. (16)

$$\mathbf{x}_k^t = \mathbf{x}_k^{t-1} + \varepsilon_{k,j} l_{ave}^{t-1} \tag{16}$$

where $\varepsilon_{k,j}$ is a uniform random number between −1 and 1 selected anew for each design variable j of the micro-bat $\mathbf{B}_k$, and l_{ave}^{t-1} is the average loudness of all micro-bats at time step $t-1$. In the proposed algorithm a reformulation of this equation is carried out for discrete structural optimization problems as proposed in Eqs. (17) and (18):

$$\varepsilon_{k,j} = N(0,\sigma) \cdot \sqrt{n_{sec}} \tag{17}$$

$$x_{k,j}^t = \begin{cases} x_{k,j}^{t-1} + round(\varepsilon_{k,j} \cdot l_{ave}^{t-1}) & : if\ u_{k,j} \geqslant r_k \\ x_{k,j}^{t-1} & : if\ u_{k,j} < r_k \end{cases} \tag{18}$$

In Eqs. (17) and (18), n_{sec} is the number of sections in the discrete set used for sizing the design variables; $x_{k,j}^t$ and $x_{k,j}^{t-1}$ are the values of jth design variable in the micro-bat $\mathbf{B}_k$ at time steps t and $t-1$, respectively; r_k is the pulse rate of the micro-bat $\mathbf{B}_k$, and $N(0,\sigma)$ is a normally distributed random number with mean 0 and standard deviation σ. The rationale behind using a normal distribution in Eq. (17) is to facilitate occurrences of small step sizes as compared to large ones during local search. Besides, the term $\sqrt{n_{sec}}$ in this equation is used to adjust 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. (16) where all design variables are subjected to transition (perturbation) during local search, Eq. (18) 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*: Once generated, the candidate micro-bats are analyzed with the set of steel sections selected for the design variables, and force and deformation responses are obtained under the loads. Objective function values of the feasible and infeasible candidate micro-bats are calculated from Eqs. (3) and (12), respectively.

Step 7. *Echolocation Parameters Update*: After evaluating candidate micro-bats, 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 from the target object. In the original BI algorithm developed by Yang [10], 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 loudness l_i of the micro-bat, then its echolocation parameters r_i, l_i are updated using the following equations:

$$l_i' = \alpha \cdot l_i \tag{19}$$

$$r_i^{t+1} = r_{max}[1 - \exp(\gamma t)] \tag{20}$$

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 proposed BI algorithm for structural optimization problems, 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. (20) is proposed for adaptation of pulse rate parameter as given in Eq. (21):

$$r_i^{t+1} = 1 - (1 - r_i^0)\gamma^{(t+1)} \leqslant r_{max} \tag{21}$$

Eq. (21) facilitates a more gradual change of pulse rate parameter from its initial (minimum) value of r_i^0 towards r_{max}, whereas in Eq. (20) the pulse rate immediately approaches r_{max} in a few iterations and remains stationary at this value thereafter. A graphical comparison of Eqs. (20) and (21) is presented in Fig. 1 by choosing r_{max} = 1.0, $r_i^0 = 0.5$, γ = 0.95 over an iteration number of 100. In the

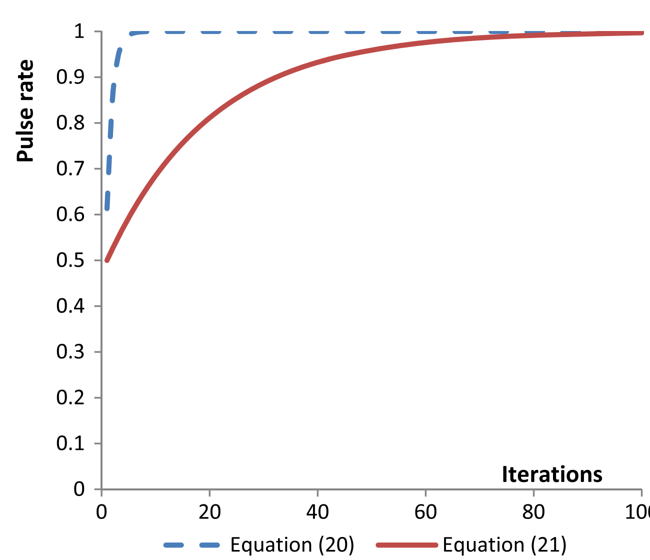


Fig. 1. Comparison of pulse rate adaptation strategies.

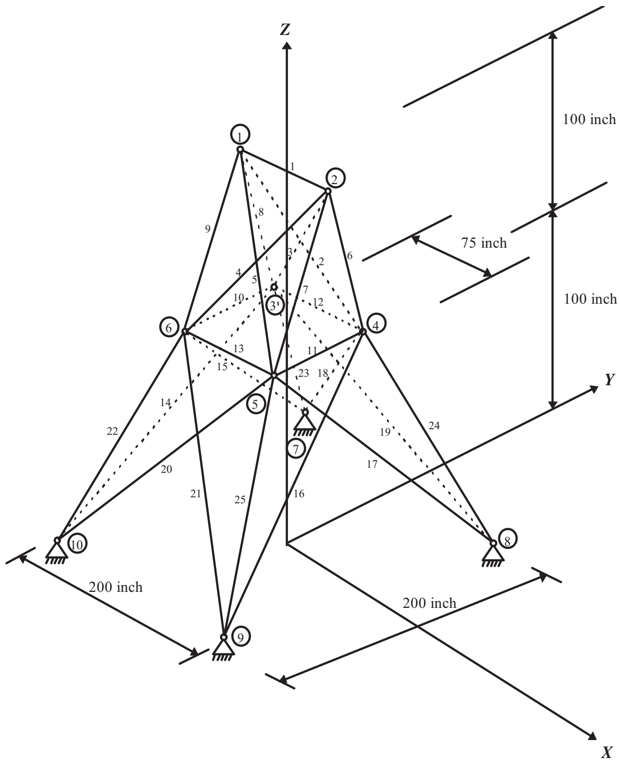


Fig. 2. 25-Member truss (benchmark problem).

numerical applications performed here, α and γ are set to 0.95 and 0.99 for all the examples, respectively.

Step 8. *Selection*: Selection is then carried out between current and candidate micro-bats to form members of the next population which will parent and guide generation of the subsequent micro-bats. Selection methodology employed in the proposed algorithm is borrowed from the well-known variant of evolution strategies technique referred to as $(\mu + \mu)$ – 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 $\mu + \mu = 2\mu$ current and candidate solutions are selected deterministically in reference to their objective function values. It should be noted that this selection methodology comes up with a promise of guaranteed survival for the micro-bats; that is to say the selected micro-bats at any iteration represent the best solutions located by the algorithm from the start.

Step 9. *Termination*: The steps 4 through 8 are implemented in the same way until a termination criterion is met, which can be imposed as a maximum number of iterations or no improvement of the best feasible design over a certain number of iterations.

4. Numerical examples

In this section performance of the proposed BI algorithm is measured using four numerical problems chosen from optimum design of pin-jointed steel trusses. These problems are (i) 25-bar truss with 8 design variables, (ii) 354-bar braced dome with 22 design variables; (iii) 693-bar braced barrel vault with 23 design variables; and finally (iv) 942-bar truss tower with 59 design variables. In all these problems the structures are sized for minimum weight by selecting the members from a set of discrete sections. The first example (25-bar truss) refers to a benchmark problem of structural size optimization that has been long used in the literature to verify new techniques or methodologies. On the other hand, the other three problems present practical design application instances of real-size problems according to provisions of AISC-ASD [14] specification. All the test problems have been studied formerly by the authors using a large set of metaheuristic techniques, including simulated annealing (SA), evolution strategies (ESs), particle swarm optimization (PSO), ant colony optimization (ACO), tabu search (TS), harmony search (HS) and simple genetic algorithms (SGA) and big bang-big crunch (BB-BC) in Refs. [15,17–20]. Therefore, comprehensive comparisons are provided between the optimum solutions obtained for these problems using the proposed BI algorithm and other metaheuristic algorithms.

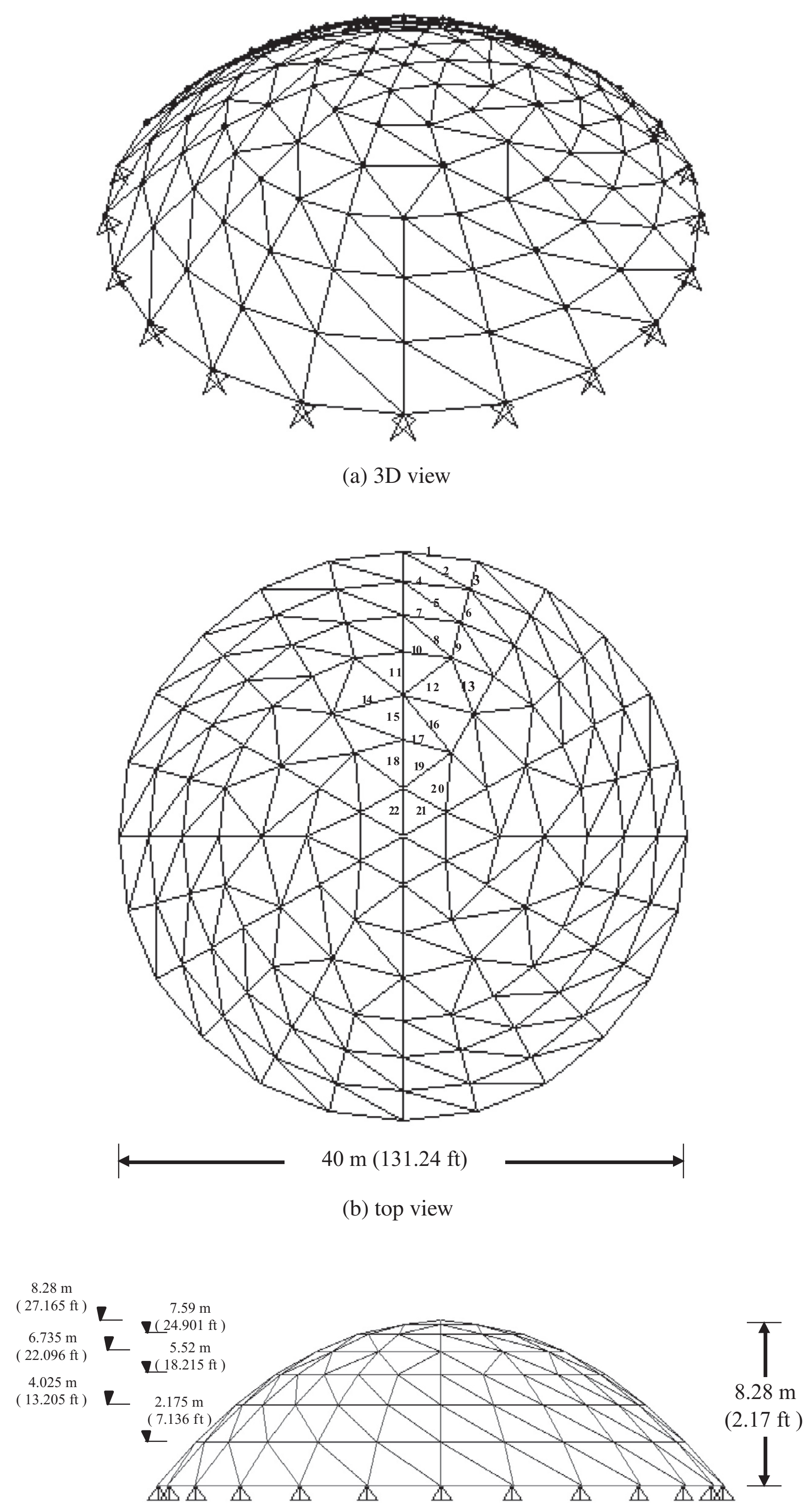
For a fair comparison of results, the maximum number of iterations is limited to 1000 for the examples 1 through 3 and 2000 for the example 4. However, in cases where no progress in the best feasible design is recorded over a certain number of successive iterations, the search process is terminated before the maximum number of structural analyses is reached. During numerical implementations, search is initiated from different starting points by generating the initial population randomly at each optimization run, and the control parameters of the BI algorithm are chosen as follows: population size $\mu = 50$, 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 $\alpha = 0.95$, initial (minimum) pulse rate $r_i^0 = r_{\min} = 0.5$, pulse rate adaptation parameter $\gamma = 0.99$, standard deviation $\sigma = 1$ for the examples 1 through 3 and $\sigma = 2.0$ for the example 4, and finally penalty coefficient $p = 1$. The material properties of steel used for practical design examples (examples 2 through 4) are taken as follows: modulus of elasticity $(E) = 29{,}000$ ksi (203,893.6 MPa) and yield stress $(F_y) = 36$ ksi (253.1 MPa).

4.1. Example 1: 25-bar truss

The standard 25-member truss shown in Fig. 2 has been frequently used in the literature for testing and comparing various optimization techniques. Two versions of the problem are available with discrete and continuous design variables. The differences

Table 1
Comparison of optimum designs for 25-member truss problem.

Sizing variable	Truss members	PSO [15]	HS [15]	SA [15]	ESs [15]	AC [15]	SGA [15]	TS [15]	BI (this study)
1	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	2–5	0.3	0.3	0.3	0.5	0.5	0.2	0.4	0.3
3	6–9	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
4	10,11	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	12,13	2.1	2.1	2.1	1.9	1.9	2.0	1.8	2.1
6	14–17	1.0	1.0	1.0	0.9	1.0	1.0	0.9	1.0
7	18–21	0.5	0.5	0.5	0.5	0.4	0.6	0.6	0.5
8	22–25	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4
Weight (lb)		484.85	484.85	484.85	485.05	485.05	485.38	485.57	484.85
Number of analyses		1,600	2,100	6,624	4,350	10,050	9,050	1,626	2,900



(a) 3D view

(b) top view

(c) side view

Fig. 3. 354-Bar braced truss dome.

between them do not only extent to type of design variables employed, but also to loading and constraints imposed on the problem. Here, the discrete version of the problem is studied owing to discrete implementation nature of the proposed BI algorithm. The 25 truss members grouped in 8 independent design variables are selected from a discrete set of 30 ready sections. A stress

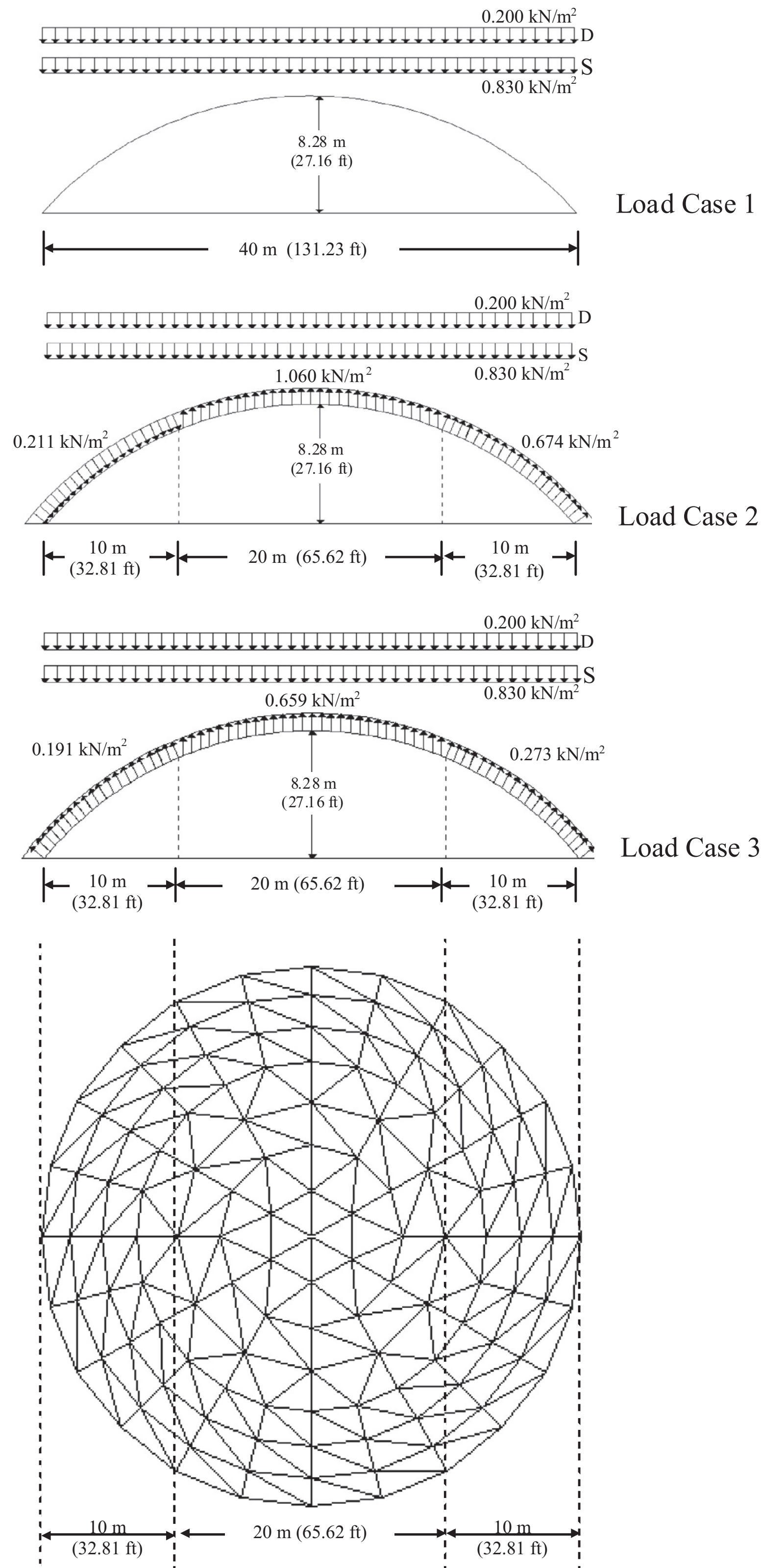


Fig. 4. The three load cases considered for 354-bar braced truss dome.

limitation of 40 ksi is imposed on members both in tension and compression, and maximum displacements of joints 1 and 2 must be less than 0.35 in along *x* and *y*-directions. Further details of the design data are readily available in the literature.

Five independent runs have been executed with the proposed BI algorithm to optimize the 25-bar truss, resulting in an optimum design weight of 484.85 lb for the truss at its best run. This solution has been located at 58th iteration, and the algo-

Table 2
Comparison of optimum designs for 354-bar braced dome.

Sizing variables	Optimal cross sectional areas (in^2)								
	SA [15]	PSO [15]	ACO [15]	TS [15]	HS [15]	SGA [15]	BB-BC [20]	BI (this study)	
								Area	Section
1	1.07	1.07	1.07	1.07	1.07	1.48	1.07	1.07	P2
2	3.17	3.17	3.17	3.17	3.17	2.68	3.17	3.17	P4
3	2.23	2.23	2.23	2.68	2.23	4.3	2.23	2.23	P3
4	2.68	2.68	2.68	2.68	2.68	2.68	2.68	2.68	P3.5
5	2.23	2.23	2.23	2.68	2.23	2.23	2.23	2.23	P3
6	2.23	2.23	2.23	2.23	2.25	2.23	2.23	2.23	P3
7	2.23	2.23	2.23	2.68	2.23	2.23	2.23	2.23	P3
8	2.23	2.23	2.23	2.68	2.68	2.23	2.23	2.23	P3
9	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	P2.5
10	2.23	2.23	2.23	2.23	2.25	2.25	2.23	2.23	P3
11	1.7	1.7	2.66	1.7	2.25	2.66	1.7	1.7	P2.5
12	1.7	1.7	2.23	1.7	1.7	2.23	1.7	1.7	P2.5
13	1.7	1.7	1.7	1.7	1.7	2.23	1.7	1.7	P2.5
14	1.7	1.7	1.7	1.7	2.23	1.7	1.7	1.7	P2.5
15	1.7	1.7	1.7	1.7	1.7	2.25	1.7	1.7	P2.5
16	1.7	1.7	1.7	2.68	1.7	1.7	1.7	1.7	P2.5
17	1.48	1.48	1.48	1.7	2.66	1.7	1.48	1.48	PX2
18	1.48	1.48	2.68	1.48	3.02	3.17	1.48	1.48	PX2
19	1.07	1.07	1.07	1.07	1.7	1.48	1.07	1.07	P2
20	1.07	1.07	1.07	1.07	1.7	1.48	1.07	1.07	P2
21	1.07	1.07	1.07	1.07	1.7	1.48	1.07	1.07	P2
22	1.07	1.07	1.48	1.07	1.07	1.7	1.07	1.07	P2
Weight, lb (kg)	32574.9 (14775.7)	32574.9 (14775.7)	33557.5 (15221.4)	35370.1 (16043.6)	34944.3 (15850.5)	36343.3 (16485)	32574.9 (14775.7)	32574.9 (14775.7)	
Number of analyses	33,370	35,950	37,050	8,602	48,300	47,550	10,100	16,850	

rithm is automatically terminated at 108th iteration since no improvement of the best design is recorded over the following 50 iterations. Amongst the five runs implemented, mean and standard deviation of the optimized weight appear as 485.76 lb and 1.06 lb, respectively, whereas the average and standard deviation of the iteration number to reach the optimum weight are 65 and 19, respectively. In Table 1 the minimum weight design of 25-bar problem obtained by the proposed algorithm is compared to the previously reported results by Hasançebi et al. [15] using different metaheuristic techniques. It is seen from Table 1 that all the techniques perform very well locating an optimum within a close range of 484.85–485.57 lb for this small-scale benchmark problem. The best solution of the problem (i.e., 484.85 lb) has been identified by the BI algorithm in addition to other three algorithms, namely PSO, HS and SA. In the literature, a plenty of different solutions of the problem ranging between 484.50 and 546.01 lb are reported with different numerical techniques [21–26]. A lighter design weight of 481.3 lb is reported with simulated annealing (SA) in Bennage and Dhingra [27], yet the authors note that this design slightly violates the displacement limitations such that joint 2 deflects 0.3514 in along y-direction.

4.2. Example 2: 354-bar braced dome truss

The second test problem is a 354-bar braced dome truss shown in Fig. 3, which has been formerly studied with SA, PSO, ACO, TS, HS, SGA in Hasançebi et al. [15] and with BB-BC technique in Hasançebi and Kazemzadeh Azad [20]. The dome has a 40 m (131.23 ft) diameter designed for covering the top of an auditorium at an elevation of 10 m (32.8 ft). It has a height of 8.28 m (27.17 ft), and consists of 127 joints and 354 members. The 354 members are grouped into 22 independent sizing variables (Fig. 3), which are selected from the entire set of 37 standard circular hollow sections. The dome is subjected to the following three load cases considering various combinations of dead (D), snow (S) and wind (W) loads computed according to the specifications of ASCE7-05 [28]: (i) D + S, (ii) D + S + W (with negative internal pressure), and (iii) D + S + W (with positive internal pressure). It is important to note that the load cases resulting from unbalanced snow loads are disregarded in the study to avoid excessive computational burden. The illustrations of the three load cases are shown in Fig. 4. It has been assumed that dead and snow loads act on the projected area, while wind load acts on the curved surface area. Sandwich type aluminum cladding material is used, resulting in an assumed dead load pressure of 0.2 kN/m^2 including the frame elements used for the girds. The stress and stability constraints of the members are computed according to the specifications of AISC-ASD [14]. The displacements of all nodes are limited to 11.1 cm (4.37 in) in any direction.

Four independent runs have been executed with the proposed BI algorithm to optimize the 354-bar braced dome truss, resulting in an optimum design weight of 14775.7 kg (32574.9 lb) at its best run. This design has been obtained at 337th iteration and the algorithm is automatically terminated at 437th iteration since no improvement of the best design is achieved over the following 100 iterations. Since all the four runs have identified the same design, mean and standard deviation of the optimized weight are 14775.7 kg and zero, respectively, whereas the average and standard deviation of the iteration number to reach this design are 361 and 20, respectively. This design is compared with literature in Table 2 in terms of discrete sections adopted for each member group and the resulting design weights. It is noted that the results published formerly by Hasançebi et al. [15] are corrected here due to mis-grouping of one bracing member in the previous work. It can be seen from Table 2 that the optimum design found by the proposed BI algorithm is overall the best solution for this test problem and coincides with that found by SA, PSO and BB-BC. Remarkably heavier structures were designed by ACO, TS, HS and SGA.

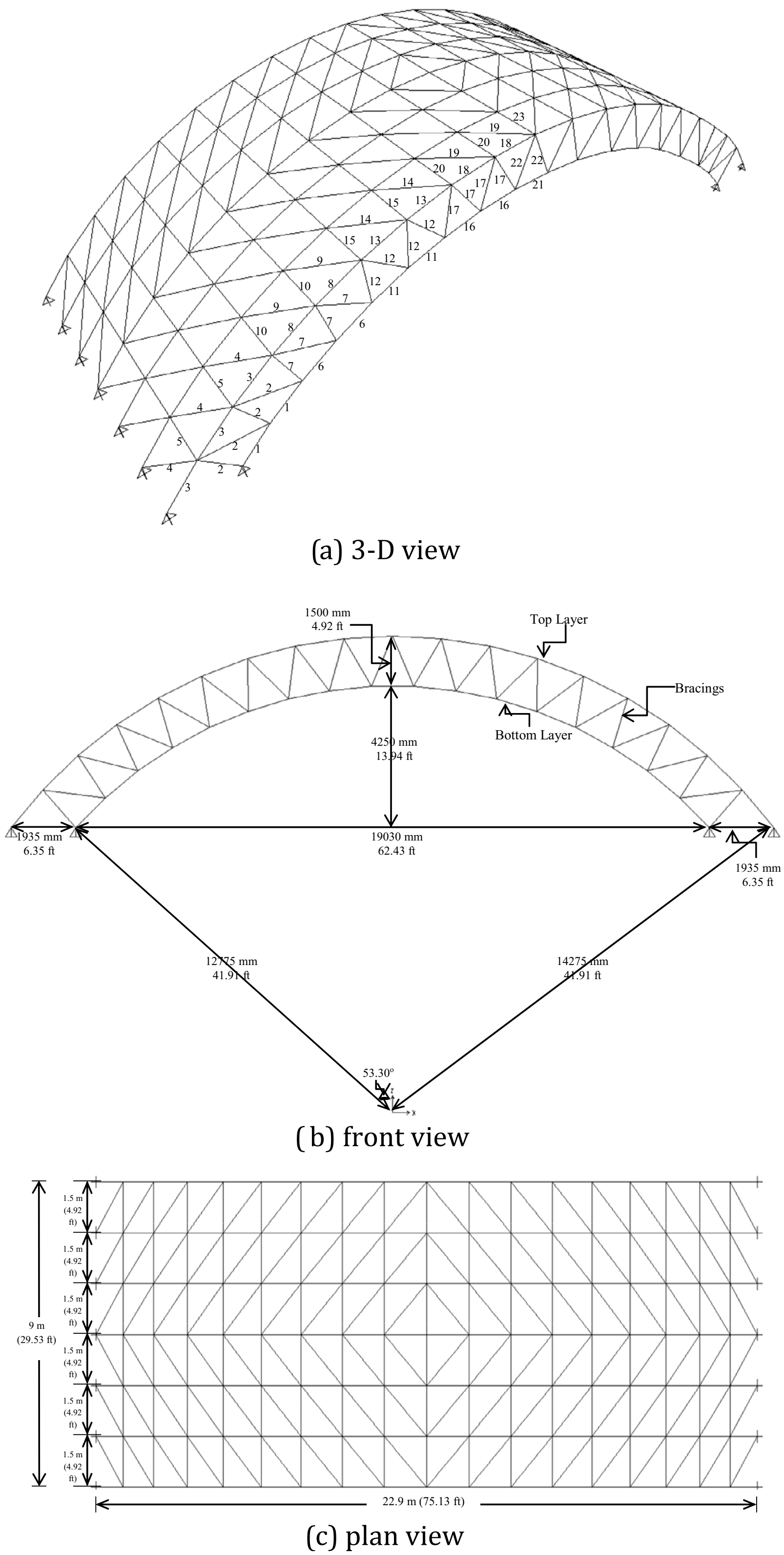


Fig. 5. 693-Bar braced barrel vault.

4.3. Example 3: 693-bar braced barrel vault

The third test problem regards the optimum design of the spatial braced barrel vault [29] shown in Fig. 5. The structure includes 259 joints and 693 members that are grouped into 23 independent sizing variables due to symmetry of the structure about the centerline. The member grouping scheme is given in Fig. 5(a) and the main geometric dimensions of the structure are shown in Fig. 5(b) and (c). It is assumed that the barrel vault is subjected to a uniform dead load (DL) pressure of 35 kg/m^2, a positive wind

Table 3
Comparison of optimum designs for 693-bar braced barrel vault.

Sizing variables	Optimal cross sectional areas (in^2)					
	HS [19]	GA [19]	ACO [19]	BB-BC [20]	BI (this study)	
					Area	Section
1	3.68	3.02	4.03	3.68	3.68	PX3.5
2	0.433	0.669	0.494	0.494	0.494	P1
3	0.494	0.639	0.494	0.333	0.333	P.75
4	0.494	0.494	0.494	0.494	0.494	P1
5	0.433	0.333	0.494	0.333	0.333	P.75
6	3.17	4.41	0.333	3.68	3.68	PX3.5
7	0.669	0.639	0.639	0.494	0.494	P1
8	0.333	0.333	0.333	0.494	0.494	P1
9	2.68	2.66	2.68	0.494	0.494	P1
10	0.494	0.639	4.03	0.333	0.333	P.75
11	0.669	0.669	0.494	2.23	2.66	PXX2
12	0.881	0.799	0.639	0.799	0.799	P1.5
13	1.07	1.07	0.881	1.07	1.07	P2
14	0.881	0.799	0.639	0.494	0.494	P1
15	0.333	0.494	0.333	0.333	0.333	P.75
16	0.881	0.669	0.639	0.881	1.07	PX1.5
17	0.881	1.07	0.881	0.669	0.639	PX1
18	0.669	0.799	0.494	2.23	1.7	P2.5
19	0.639	0.669	0.669	0.494	0.494	P1
20	0.333	0.333	0.333	0.333	0.333	P.75
21	1.7	2.23	1.7	0.494	0.494	P1
22	0.494	0.669	0.669	0.333	0.333	P.75
23	0.639	0.433	0.494	0.333	0.333	P.75
Weight, lb (kg)	11232.71 (5095.07)	12029.49 (5456.48)	10999.20 (4989.15)	10595.33 (4805.96)	10564.84 (4792.13)	
Number of analyses	48,150	47,250	45,650	NA	36,300	

load (WL) pressure of 160 kg/m^2 (32.77 lb/ft^2) and a negative wind load (WL) pressure of 240 kg/m^2 (49.16 lb/ft^2). Here, these loads are combined under two separate load cases as follows: (i) 1.5DL + 1.5WL = 1.5(35 + 160) = +292.5 kg/m^2 (59.91 lb/ft^2) and (ii) 1.5DL − 1.5WL = 1.5(35 − 240) = −307.5 kg/m^2 (62.98 lb/ft^2), along z-direction. The displacements of all joints in x, y, and z directions must be less than 0.254 cm (0.1 in). The strength and stability requirements of steel members are imposed according to AISC-ASD [14]. Structural members are selected from the entire list of 37 standard circular hollow sections.

The 693-bar braced barrel vault has been previously studied with SGA, HS and ACO techniques in Hasançebi et al. [19] and with BB-BC technique in Hasançebi and Kazemzadeh Azad [20]. In this study five independent runs have been executed with the proposed BI algorithm to optimize the barrel vault, resulting in an optimum design weight of 4792.13 kg (10564.84 lb) for the truss at its best run. This design has been obtained at 726th iteration and the algorithm is automatically terminated at 826th iteration since no improvement of the best design is achieved over the following 100 iterations. Amongst the five runs implemented, mean and standard deviation of the optimized weight appear as 4806.11 kg and 11.12 lb, respectively, whereas average and standard deviation of the iteration number to reach the optimum weight are 721 and 190, respectively. The optimum design found by the proposed BI algorithm is compared with literature in Table 3 in terms of discrete cross-sections selected by the optimizer for each member group and resulting structural weight. It can be seen that BI is the most efficient meta-heuristic optimizer overall as it converges to the best design obtained so far.

4.4. Example 4: 942-bar truss tower

The last test problem regards the 942-bar truss tower shown in Fig. 6. Symmetry of the tower around x and y-axes is employed to group the 942 truss members into 59 independent sizing variables. The truss members are selected from W-shape profile list consisting of 295 discrete sections. The tower is subject to a single loading condition consisting of both horizontal and vertical loads, as follows: (i) the vertical loads in the z direction are −3.0 kips (−13.344 kN), −6.0 kips (−26.688 kN) and −9.0 kips (−40.032 kN) at each node in the first, second and third sections, respectively; (ii) the lateral loads in the y direction are 1.0 kips (4.448 kN) at all nodes of the tower; and (iii) the lateral loads in the x direction are 1.5 kips (6.672 kN) and 1.0 kips (4.448 kN) at each node on the left and right sides of the tower, respectively. The stress and stability constraints are imposed according to the provisions of AISC-ASD [14] specification, and the displacements of all nodes in any direction must be less than 15 in (38.1 cm).

This problem was first studied in Hasançebi and Erbatur [17] using SA: the corresponding optimum weight of the tower was 172,214 kg (379,660 lb). Later the same problem has been handled in Hasançebi [18] using two population-based evolutionary methods, where the best design weights of 178,864 kg (394,321 lb) and 172,214 kg (379,660 lb) were located for this tower using SGA and ESs techniques, respectively. In the present study six independent runs have been executed with the proposed BI algorithm to optimize the tower, resulting in an optimum design weight of 171,261 kg (377,567 lb) for the tower at its best run. This design has been obtained at 1914th iteration and the algorithm is terminated when all iterations are completed. Amongst the six runs implemented, mean and standard deviation of the optimized weight appear as 173,283 kg and 1801 kg, respectively, whereas average and standard deviation of the iteration number to reach the optimum weight are 1896 and 68, respectively. The optimum design found by the proposed BI algorithm is compared with literature in Table 4 in terms of discrete cross-sections selected by the optimizer for each member group and resulting structural weight. It can be seen that BI is the most efficient meta-heuristic optimizer overall as it converges to the best design obtained so far.

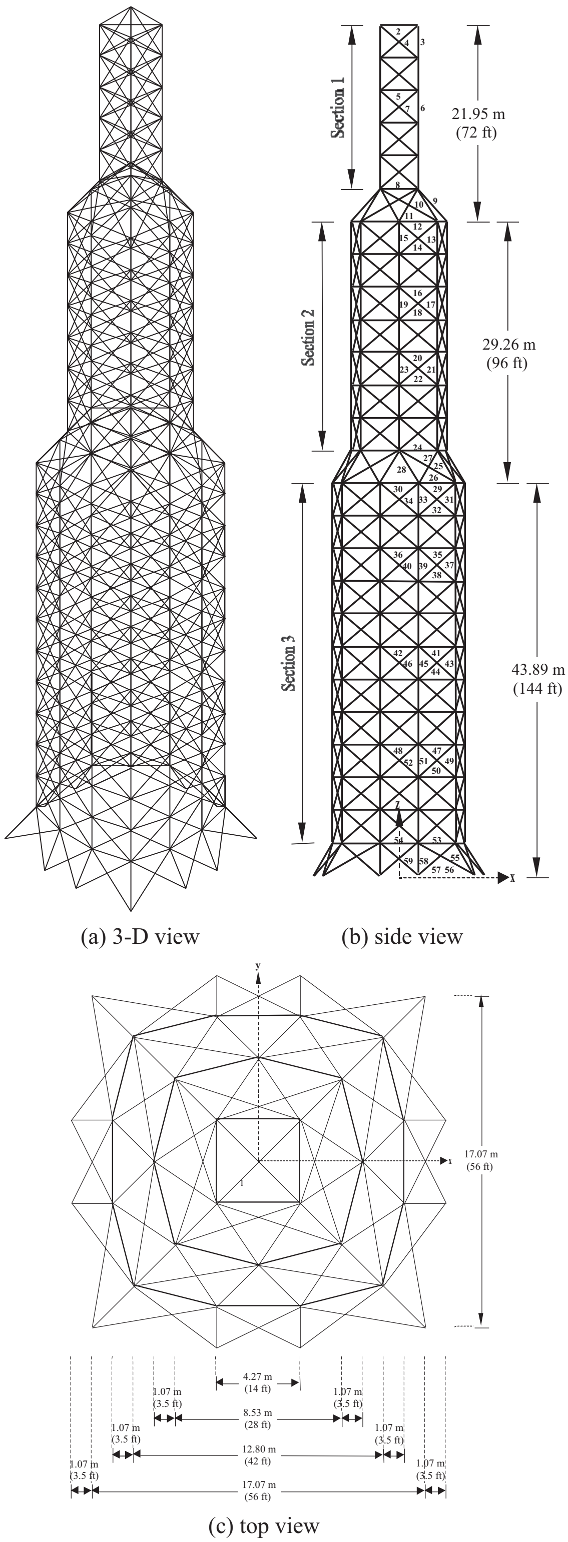


Fig. 6. 26-Story, 942-bar space truss tower.

5. Discussions on parameters role and sensitivity

The BI algorithm developed in the study includes a considerable amount of heuristics and depends on many echolocation parameters. In this section we intend to provide a detailed insight into roles and functions of these parameters in a search process. Besides, sensitivity of optimum design to each parameter is

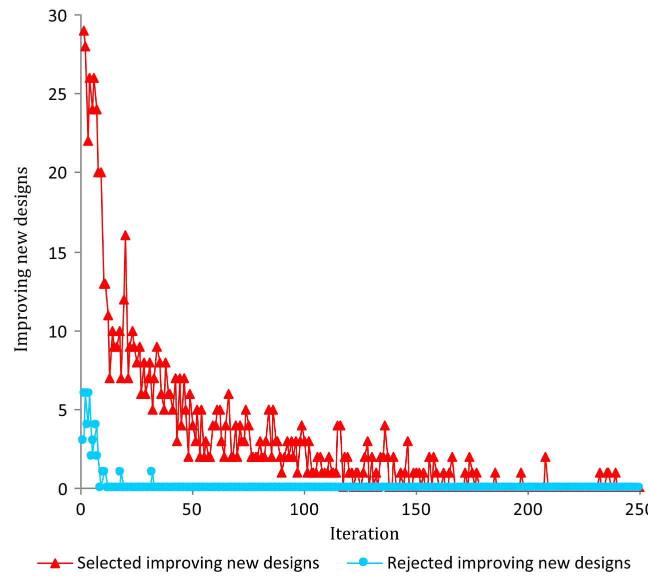


Fig. 7. Number of improving new designs selected and rejected at each iteration in a typical optimization run of 25-bar truss problem.

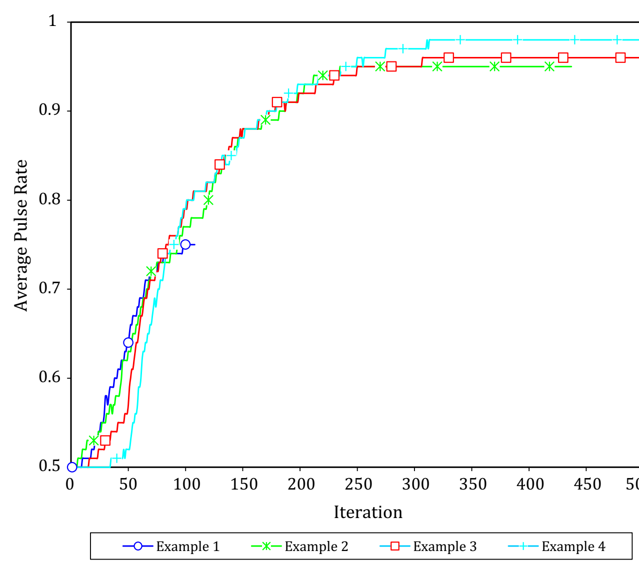


Fig. 8. Variation of average pulse rate parameter in test problems.

explicated based on extensive numerical experiments performed during the design examples.

In the proposed algorithm lower and upper bounds are imposed on the echolocation parameters, as discussed in Section 3. This is essential not only to accommodate better moves in the design space, but also to maintain search efficiency of the algorithm while converging towards the optimum. In this regard the lower and upper bounds ($f_{\min}$ and $f_{\max}$) associated with the frequency parameter ensure that a coherent velocity is allocated to micro-bats during random flying. This should be clear from Eq. (14), where the change in the velocity of a micro-bat in a certain direction is set to a value between zero and $\mathbf{x}_i^{t-1} - \mathbf{x}_*$ when $f_{\min}$ and $f_{\max}$ are chosen as zero and one, respectively. The idea here is to bring the micro-bat to a new point between its current position and global best design, assuming that better solutions might exist between these two favorable design points. When a larger value is used for $f_{\max}$,

Table 4
Comparison of optimum designs for 942-bar space truss.

Sizing variables	Optimal cross sectional areas section (area, in^2)			
	SA [17]	SGA [18]	ESs [18]	BI (this study)
1	W6 × 9 (2.68)	W10 × 22 (6.49)	W6 × 9 (2.68)	W6 × 9 (2.68)
2	W6 × 9 (2.68)	W6 × 9 (2.68)	W8 × 10 (2.96)	W6 × 9 (2.68)
3	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
4	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
5	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
6	W6 × 15 (4.43)	W5 × 19 (5.54)	W6 × 15 (4.43)	W6 × 15 (4.43)
7	W6 × 15 (4.43)	W5 × 16 (4.68)	W6 × 15 (4.43)	W6 × 15 (4.43)
8	W6 × 9 (2.68)	W14 × 22 (6.49)	W6 × 9 (2.68)	W6 × 9 (2.68)
9	W6 × 20 (5.87)	W18 × 50 (14.70)	W6 × 20 (5.87)	W6 × 20 (5.87)
10	W8 × 24 (7.08)	W8 × 24 (7.08)	W6 × 25 (7.34)	W8 × 24 (7.08)
11	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
12	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
13	W6 × 20 (5.87)	W10 × 22 (6.49)	W6 × 20 (5.87)	W6 × 20 (5.87)
14	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
15	W4 × 13 (3.83)	W5 × 16 (4.68)	W4 × 13 (3.83)	W4 × 13 (3.83)
16	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
17	W8 × 28 (8.25)	W8 × 28 (8.25)	W8 × 28 (8.25)	W8 × 28 (8.25)
18	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
19	W6 × × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W5x16 (4.68)
20	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
21	W8 × 35 (10.30)	W8 × 35 (10.30)	W8 × 35 (10.30)	W8 × 35 (10.30)
22	W6 × 20 (5.87)	W6 × 20 (5.87)	W6 × 20 (5.87)	W6 × 20 (5.87)
23	W6 × 25 (7.34)	W8 × 31 (9.13)	W8 × 24 (7.08)	W8 × 24 (7.08)
24	W8 × 35 (10.30)	W12 × 40 (11.80)	W10 × 45 (13.30)	W8 × 35 (10.30)
25	W10 × 49 (14.40)	W8 × 58 (17.10)	W8 × 58 (17.10)	W10 × × 49 (14.40)
26	W8 × 31 (9.13)	W10 × 33 (9.71)	W8 × 31 (9.13)	W8 × 31 (9.13)
27	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
28	W8 × 24 (7.08)	W12 × 26 (7.65)	W8 × 24 (7.08)	W8 × 24 (7.08)
29	W14 × 26 (7.69)	W8 × 24 (7.08)	W6 × 25 (7.34)	W8 × 24 (7.08)
30	W8 × 21 (6.16)	W14 × 22 (6.49)	W10 × 22 (6.49)	W8 × 21 (6.16)
31	W12 × 87 (25.60)	W10 × 68 (20.00)	W14 × 90 (26.50)	W27 × 84 (24.80)
32	W6 × 20 (5.87)	W8 × 24 (7.08)	W6 × 20 (5.87)	W6 × 20 (5.87)
33	W6 × 20 (5.87)	W6 × 15 (4.43)	W6 × 15 (4.43)	W5x19 (5.54)
34	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
35	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
36	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
37	W14 × 99 (29.10)	W24 × 104 (30.60)	W14 × 99 (29.10)	W14 × 99 (29.10)
38	W8 × 24 (7.08)	W8 × 24 (7.08)	W8 × 24 (7.08)	W8 × 24 (7.08)
39	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
40	W6 × 20 (5.87)	W6 × 20 (5.87)	W6 × 20 (5.87)	W6 × 20 (5.87)
41	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)	W6 × 9 (2.68)
42	W6 × 9 (2.68)	W4 × 13 (3.83)	W8 × 10 (2.96)	W6 × 9 (2.68)
43	W24 × 131 (38.50)	W12 × 136 (39.90)	W24 × 131 (38.50)	W24 × 131 (38.50)
44	W8 × 31 (9.13)	W8 × 31 (9.13)	W8 × 31 (9.13)	W8 × 31 (9.13)
45	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)	W6 × 15 (4.43)
46	W8 × 24 (7.08)	W8 × 24 (7.08)	W8 × 24 (7.08)	W8 × 24 (7.08)
47	W4 × 13 (3.83)	W8 × 18 (5.26)	W4 × 13 (3.83)	W4 × 13 (3.83)
48	W6 × 9 (2.68)	W6 × 20 (5.87)	W6 × 9 (2.68)	W6 × 9 (2.68)
49	W14 × 145 (42.70)	W14 × 145 (42.70)	W14 × 145 (42.70)	W14 × 145 (42.70)
50	W8 × 31 (9.13)	W8 × 31 (9.13)	W8 × 31 (9.13)	W8 × 31 (9.13)
51	W8 × 28 (8.25)	W6 × 20 (5.87)	W12 × 30 (8.79)	W8 × 28 (8.25)
52	W8 × 24 (7.08)	W8 × 31 (9.13)	W8 × 24 (7.08)	W8 × 24 (7.08)
53	W10 × 60 (17.60)	W14 × 61 (17.90)	W12 × 65 (19.10)	W12 × 65 (19.10)
54	W24 × 68 (20.10)	W8 × 48 (14.10)	W21 × 73 (21.5)	W21 × 73 (21.5)
55	W14 × 132 (38.80)	W14 × 120 (35.30)	W14 × 132 (38.80)	W14 × 132 (38.80)
56	W8 × 35 (10.30)	W8 × 31 (9.13)	W8 × 31 (9.13)	W8 × 31 (9.13)
57	W12 × 79 (23.20)	W10 × 100 (29.40)	W12 × 72 (21.10)	W12 × 72 (21.10)
58	W8 × 24 (7.08)	W10 × 33 (9.71)	W8 × 28 (8.25)	W8 × 28 (8.25)
59	W8 × 35 (10.30)	W10 × 33 (9.71)	W8 × 31 (9.13)	W8 × 31 (9.13)
Weight, lb (kg)	379,660 lb (172,214 kg)	394,321 lb (178,864 kg)	377,947 lb (171,437 kg)	377,567 lb (171,261 kg)
Number of analyses	41,462	200,000	150,000	95,700

either micro-bats may approach the global best design very rapidly without having a chance to sample these potentially good intermediate solutions, or the search can be directed beyond the global best design due to very large step sizes occurring under high values of $f_{\max}$.

The loudness parameter controls local search features of micro-bats in the BI algorithm. The maximum loudness $l_{\max}$ is introduced to define a physical limit for extent of the region scanned by a micro-bat during local search. The numerical experiments have indicated that a sufficiently large space is provided for local search when $l_{\max}$ is chosen as 1.0. The minimum loudness defined in Eq. (10), on the other hand, avoids dropping of the average loudness parameter l_{ave} too much, in which case local search may lead to a very slow convergence or may become totally ineffective. It should be noted that in the proposed algorithm a design variable in a micro-bat is changed in an amount equal to the rounded value of

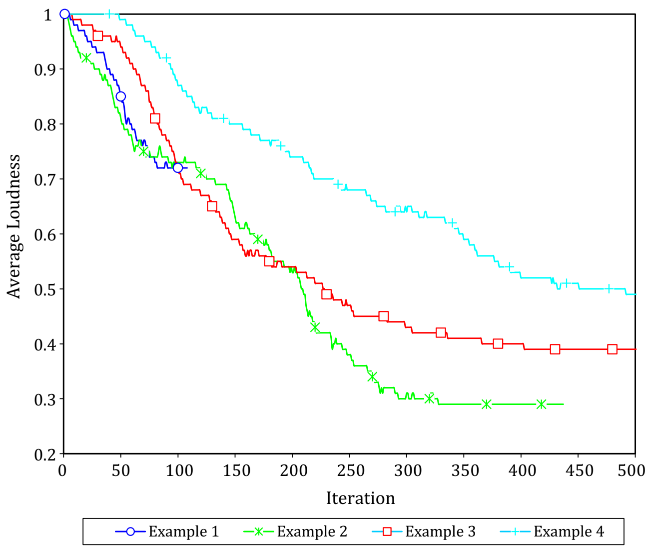


Fig. 9. Variation of average loudness parameter in test problems.

$\Delta x = N(0,\sigma) \cdot \sqrt{n_{sec}} \cdot l_{ave}$ by virtue of Eqs. (18) and (19). Substituting the lowest possible value of l_{ave} in this equation, (i.e., $l_{min} = 1/\sqrt{n_{sec}}$) yields $\Delta x = N(0,\sigma)$, indicating that minimal variation of a design variable during local search is controlled by a normally distributed number. The numerical experiments have indicated that 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 pulse rate parameter has dual roles in the BI algorithm. On the one hand it decides upon the search scheme followed by a micro-bat (i.e., random flying or local search), and on the other hand it controls the number of design variables changed in a micro-bat during local search. The initial choice of this parameter specifies its lower bound r_{min} as well, and it is arbitrarily set to $r_{min} = 0.5$ in the present study to place identical emphasis on global and local searches in the beginning of the optimization process. This parameter can be set to a smaller value if more explorative search is required initially or vice versa. The upper bound $r_{max} = 1 - 1/N_d$ serves two tasks. Firstly, although local search is encouraged progressively over time, the use of r_{max} ensures that random flying which characterizes a typical global search mechanism is not totally abandoned by all micro-bats. Secondly, considering the fact that the probability for change of a design variable is set to $1 - r$ by Eq. (18), it is assured that at least one design variable is changed probabilistically to generate dissimilar design points during local search.

As mentioned before, echolocation parameters are modified through the adaptation parameters in favor of a more effective search during the optimization process. It has been found that a suitable value for loudness adaptation parameter α for truss sizing optimization problems is 0.95, yet the recommended value set of this parameter lies in the range [0.95, 0.99]. In general it should be noted that the larger this parameter is, more slowly does the loudness decrease. Accordingly, by virtue of Eqs. (16)–(19) local search is performed over wide regions even at late iterations. This may slow down convergence of the algorithm for relatively small and regular design spaces, yet it is very beneficial for large and irregular design spaces for which escaping from local optima is not easy. Hence, this parameter can be set to a value above 0.95 for problems having a large and irregular fitness landscape. The results have also indicated that the optimal value for pulse rate adaptation parameter γ is 0.99 when the algorithm is executed over a maximum iteration number of 1000. This parameter can be set to a smaller value if lesser iterations are performed.

As mentioned before the proposed BI algorithm employs a $(\mu + \mu)$ selection strategy that is borrowed from evolution strategies. This selection methodology comes up with an elitist strategy such that the overall μ best individuals are selected to survive in the next iteration. This implies that any new trial design corresponding to the updated position of a micro-bat may not necessarily be included in the new population even though it might have improved the design with respect to its current best position. This issue has been investigated numerically with respect to the 25-bar truss problem. In a typical run of 25-bar truss problem the number of improving new designs selected and rejected at each iteration of a $(\mu + \mu)$ selection strategy is displayed in Fig. 7. It is observed from this figure that rejection of improving designs occasionally takes place in the very early iterations only. After about 30th iteration number, all the improving designs are accepted to the population. In fact, a total of 805 improving designs are generated throughout the optimization process and 771 out of them are accepted, resulting in 96% acception rate. This ratio confirms efficiency of a $(\mu + \mu)$ selection strategy in the proposed BI algorithm.

Finally, it is important to stress out that stagnation of the BI algorithm in local optima is prevented by means of two parameters r_{max} and l_{min}. The use of r_{max} as an upper value of pulse rate parameter ensures that a few micro-bats are always encouraged to carry on their search with random flying mechanism in case the algorithm may get stuck in local optima. In addition, the size of region scanned through local search is always kept sufficiently large through the l_{min} to avoid entrapment of the search in local optima.

6. Conclusions

In this study a novel optimization algorithm is developed as an effective method for structural optimization problems 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 one benchmark and three practical design examples with medium to large scales. In these test problems performance of the proposed BI algorithm is measured against a variety of different metaheuristic techniques under the same design considerations and the optimum solutions attained by them are compared extensively in Tables 1–4. Optimization results clearly demonstrate the efficiency of the proposed algorithm which found the best known design for the first two test problems and converged to improved designs in the last two test problems. The results also show that the BI algorithm has also a favorable convergence speed compared to the most of other metaheurictic techniques, yet there are some algorithms which converge to the optimum or a near-optimum solution faster in some problems; such as BB-BC algorithm in 354-member braced dome truss and SA algorithm in 942-bar truss tower problems.

Apparently, the robustness of the BI 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. The variations of pulse rate and loudness parameters during the search process are presented in Figs. 8 and 9, respectively, in the best runs of the algorithm for the four test problems. The plots confirm the efficiency of the search process carried out by the proposed BI algorithm.

References

[1] Holland JH. Adaptation in natural and artificial systems. Arbor, MI: The University of Michigan Press; 1975.
[2] Goldberg DE, Samtani MP. Engineering optimization via genetic algorithm. In: Proc. of the 9th Conf. on Electronic Computation. ASCE; 1986. p. 471–82.
[3] Schwefel H-P. Numerical optimization of computer models. Chichester, UK: John Wiley & Sons; 1981.
[4] Kirkpatrick S, Gerlatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–80.
[5] Kennedy J, Eberhart R. Particle swarm optimization. IEEE International Conference on Neural Networks, vol. 4. IEEE Press; 1995. p. 1942–8.
[6] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colony. In: Proc. of the first European conf. on artificial life, USA; 1991. p. 134–42.
[7] Lamberti L, Pappalettere C. Metaheuristic design optimization of skeletal structures: a review. In: Topping BHV, editor. Computational technology reviews, vol. 4; 2011. p. 1–32. doi: http://dx.doi.org/10.4203/CTR.4.1.
[8] Saka MP. Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. In: Topping BHV, editor. Civil engineering computations: tools and techniques. Stirlingshire, UK: Saxe-Coburg Publications; 2007. p. 105–47. http://dx.doi.org/10.4203/CSETS.16.6.
[9] Saka MP, Doğan E. Recent developments in metaheuristic algorithms: a review. In: Topping BHV, editor. Computational technology reviews, vol. 5; 2012. p. 31–78. doi: http://dx.doi.org/10.4203/CTR.5.2.
[10] Yang X-S. A new metaheuristic bat-Inspired algorithm. In: Gonzalez JR et al., editors. Nature inspired cooperative strategies for optimization (NISCO 2010). Studies in computational intelligence. Berlin: Springer; 2010. p. 65–74.
[11] Yang X-S. Bat algorithm for multiobjective optimization. Int J Bio-Inspired Comput 2011;3:267–74.
[12] Yang X-S, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization. Eng Comput 2012;29:464–83.
[13] Gandomi AH, Yang X-S, Alavi AH, Talatahari S. Bat algorithm for constrained optimization tasks. Neural Comput Appl 2012. http://dx.doi.org/10.1007/s00521-012-1028-9.
[14] AISC-ASD, Manual of steel construction-allowable stress design. 9th ed., Chicago, IL, USA; 1989.
[15] Hasançebi O, Çarbas S, Doğan E, Erdal F, Saka MP. Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Comput Struct 2009;87:284–302.
[16] Bäck T. Evolutionary algorithms in theory and practice. New York: Oxford University Press; 1996.
[17] Hasançebi O, Erbatur F. On efficient use of simulated annealing in complex structural optimization problems. Acta Mech 2002;157:27–50.
[18] Hasançebi O. Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. Comput Struct 2008;86:119–32.
[19] Hasançebi O, Çarbas S, Saka MP. A reformulation of the ant colony optimization algorithm for large scale structural optimization. In: Tsompanakis Y, Topping BHV, editors. Proc. of the second int. conference on soft computing technology in civil, structural and environmental engineering. Stirlingshire, UK: Civil-Comp Press; 2011. http://dx.doi.org/10.4203/CCP.97.12 [Paper 12].
[20] Hasançebi O, Kazemzadeh Azad S. Discrete size optimization of steel trusses using a refined big bang–big crunch algorithm. Eng Optim 2012. http://dx.doi.org/10.1080/0305215X. 2012.748047.
[21] Rajeev S, Krishnamoorty CS. Discrete optimization of structures using genetic algorithms. J Struct Eng – ASCE 1992;118:1233–50.
[22] Erbatur F, Hasançebi O, Tütüncü İ, Kılıç H. Optimal design of planar and space structures with genetic algorithms. Comput Struct 2000;75:209–24.
[23] Ponteresso P, Fox DSJ. Heuristically seeded genetic algorithms applied to truss optimisation. Eng Comput 1999;15:345–54.
[24] Cao G. Optimized design of framed structures using a genetic algorithm. PhD thesis, Univ. of Memphis; 1996.
[25] Camp CV, Bichon BJ. Design of space trusses using ant colony optimization. J Struct Eng – ASCE 2004;130:741–51.
[26] Camp CV. Design of space trusses using big bang–big crunch optimization. J Struct Eng – ASCE 2007;133:999–1008.
[27] Bennage WA, Dhingra AK. Single and multi-objective structural optimization in discrete-continuous variables using simulated annealing. Int J Numer Methods Eng 1995;38:2753–73.
[28] ASCE 7-05, Minimum design loads for building and other structures. ASCE; 2005.
[29] Ramaswamy GS, Eekhout M, Suresh GR. Analysis, design and construction of steel space frames. Thomas Telford Publishing; 2002.