

## OPTIMAL DESIGN OF TRUSS STRUCTURES BY IMPROVED MULTI-OBJECTIVE FIREFLY AND BAT ALGORITHMS

S. Gholizadeh<sup>\*†</sup> H. Asadi and A. Baghchevan  
*Department of Civil Engineering, Urmia University, Urmia, Iran*

### ABSTRACT

The main aim of the present paper is to propose advanced multi-objective optimization algorithms (MOOAs) to tackle truss structure optimization problems. The proposed meta-heuristic algorithms are based on the firefly algorithm (FA) and bat algorithm (BA), which have been recently developed for single-objective optimization. In order to produce a well distributed Pareto front, some improvements are implemented on the basic algorithms. The proposed MOOAs are examined for three truss optimization problems, and the results are compared to those of some other well-known methods. The numerical results demonstrate that the proposed MOOAs possess better computational performance compared to the other algorithms.

Received: 20 May 2014; Accepted: 12 October 2014

**KEY WORDS:** multi-objective optimization; pareto front; firefly algorithm; bat algorithm; truss.

### 1. INTRODUCTION

The main aim of structural optimization problems is to minimize a function of structural design variables, such as the weight or cost of the structure, subject to some constraints on stresses, displacements, frequencies and so on. In fact, optimization is the process of searching for a solution such that no other superior solution can be found. In the most real-world problems, multiple conflicting objectives must be satisfied simultaneously in order to obtain optimal solutions. For example, for optimization of a truss structure the following objectives can be considered: minimizing the total weight, minimizing maximum deflection,

---

<sup>\*</sup>Corresponding author: Department of Civil Engineering, Urmia University, Urmia, Iran

<sup>†</sup>E-mail address: s.gholizadeh@urmia.ac.ir (S. Gholizadeh)

maximizing the allowable stress of members and so on. Formulation of multi-objective optimization lets such multiple objectives to be handled in the framework of an optimization problem. By solving a multi-objective optimization problem (MOOP) a set of solutions, which provide valuable information about the design problem at hand, are obtained. Investigating the obtained solutions in various viewpoints enables the designers to select their desired solution among all the solutions.

During the last years, a number of multi-objective optimization algorithms (MOOA) have been proposed by researchers, such as non-dominated sorting GA (NSGA-II) [1], Pareto archive evolution strategy (PAES) [2] and multi-objective particle swarm optimization (MOPSO) [3]. Up to now, many successful applications of various MOOAs have been reported in the literature to tackle the structural optimization problems. Coello and Christiansen [4] utilized genetic algorithm (GA) for solving MOOP of truss structures. In their study the weight of the truss, the displacement of each free node and the stress that each member has to support were minimized. Luh and Chueh [5] proposed an algorithm for finding constrained Pareto-optimal solutions based on the features of a biological immune system. The objective of their study is to minimize the weight of trusses and the maximum displacement at a certain node. Kaveh et al. [6] used GA for performing optimal design of reinforced concrete retaining walls considering minimization of the economic cost and reinforcing bar congestion as the objective functions. In [7] Kaveh and Laknejadi designed truss structures by a MOOA based on a modified multi-objective particle swarm optimization, tournament decision making process, and a local search algorithm. As well as the other reviewed works, their objective functions are the weight of structures and the maximum displacement of a certain node in a specific direction. Richardson et al. [8] integrated GA and kinematic stability repair (KSR) strategy for single and multi-objective topology optimization of truss-like structures. In a most recent work, Kaveh and Laknejadi [9] proposed a MOOA based on charged system search (CSS) meta-heuristic. They employed the CSS as an optimizer in combination with clustering and particle regeneration procedures.

In the present study, two new MOOA are proposed in which firefly algorithm (FA) [10] and bat algorithm (BA) [11] acts as the main optimization engines. Three illustrative examples of truss optimization are presented to demonstrate the efficiency of the proposed MOOAs.

## 2. FORMULATION OF MOOPS

In fact solving a MOOP is the process of finding a vector of design variables to minimize a vector function satisfying some constraints. This means that MOOPs are more complicated compared with single objective optimization and instead of finding a single solution an optimality front must be determined. In order to formulate MOOPs some basic concepts [12] can be described as follows:

**General Multi-Objective Optimization Problem:** General form of a MOOP can be stated as follows:

$$\begin{aligned} \text{Find: } X &= \{x_1, x_2, \dots, x_n\}^T, X \in \Delta & (1) \\ \text{To minimize: } F(X) &= \{f_1(X), f_2(X), \dots, f_m(X)\}^T & (2) \\ \text{Subject to: } g_i(X) &\leq 0, i = 1, 2, \dots, k & (3) \end{aligned}$$

where  $X$  is the vector of design variables of size  $n$ ;  $\Delta$  is the domain of the design variables;  $F$  is the vector function of  $m$  objective functions;  $g_i$  is the  $i$ th constraint and  $k$  is the number of constraints.

**Pareto Dominance:** Vector  $Z^1 = \{z_1^1, z_2^1, \dots, z_n^1\}^T$  dominates vector  $Z^2 = \{z_1^2, z_2^2, \dots, z_n^2\}^T$  if and only if  $Z^1$  is partially less than  $Z^2$ . This statement can be mathematically represented as follows:

$$Z^1 < Z^2 \text{ if } (\forall i \in \{1, 2, \dots, n\}, z_i^1 \leq z_i^2 \wedge \exists i \in \{1, 2, \dots, n\}: z_i^1 < z_i^2) \quad (4)$$

**Pareto Optimal:** A candidate solution  $X \in \Delta$  is Pareto optimal if and only if there is no other solution  $\bar{X} \in \Delta$  for which  $Z^2 = \{f_1(\bar{X}), f_2(\bar{X}), \dots, f_m(\bar{X})\}^T$  dominates  $Z^1 = \{f_1(X), f_2(X), \dots, f_m(X)\}^T$ . The phrase Pareto optimal is taken to mean with respect to the entire design variable space unless otherwise specified.

**Pareto Optimal Set:** Pareto optimal or non-dominated solution is defined as a solution that it is not dominated by another solution. The Pareto optimal set  $P$  for a MOOP can be defined with respect to the vector function  $F(X)$  as follows:

$$P = \{X \in \Delta | \exists \bar{X} \in \Delta: F(\bar{X}) < F(X)\} \quad (5)$$

**Pareto Front:** The Pareto front (PF) for a MOOP can be defined with respect to the vector function  $F(X)$  and Pareto optimal set  $P$  as follows:

$$PF = \{F(X) = \{f_1(X), f_2(X), \dots, f_m(X)\}^T | X \in P\} \quad (6)$$

In general, for complex MOOPs, finding exact PF is not possible and in such cases, the aim is to find a Pareto optimal set that approximates the exact PF as close as possible by generating a diverse range of solutions.

### 3. METAHEURISTIC ALGORITHMS

Metaheuristics have received more and more popularity in the last years. Their use in many applications shows their efficiency and effectiveness to solve large and complex problems [13]. During the recent years, many Metaheuristics have been developed by researchers and the most popular ones are GA, PSO, CSS, harmony search (HS), ant colony optimization (ACO), and etc. In the field of structural optimization, many successful applications of these algorithms have been reported in literature. Firefly algorithm (FA) and bat algorithm (BA) are the recent additions to the metaheuristics and their superiority to GA and PSO for handling engineering and structural optimization problems have been demonstrated in [14-15].

A glance at the literature on the multi-objective optimization reveals that the well-known

GA, PSO and CSS metaheuristics were used by researchers as the main optimizer in the framework of MOOAs. A simple version of multi-objective firefly algorithm (MOFA) has been proposed in [16] for solving continuous optimization problems. Also, in [17] BA has been used to design a simple MOOA based on combination of all objectives into a single function. It is clear that optimization of a weighted sum of all objectives as a single objective is inadequate for multi-modal problems with large number of variables. In the present paper two new and more efficient MOOAs based on FA and BA are proposed to tackle structural optimization problems. The next sections describe the basic concepts of FA and BA and their multi-objective versions.

### 3.1 Standard firefly algorithm

The FA is a metaheuristic inspired by the flashing behaviour of fireflies. The FA is a population-based algorithm, which may share many similarities with PSO. Fireflies communicate, search for prey and find mates using bioluminescence with varied flashing patterns. In order to develop the firefly algorithm, natural flashing characteristics of fireflies have been idealized using the following three rules [10]:

All of the fireflies are unisex; thus, one firefly will be attracted to other ones regardless of their sex.

Attractiveness of each firefly is proportional to its brightness, thus for any two flashing fireflies, the less bright firefly will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.

The brightness of a firefly is determined according to the nature of the objective function.

The attractiveness of a firefly is determined by its brightness or light intensity which is obtained from the objective function of the optimization problem. However, the attractiveness  $\beta$ , which is related to the judgment of the beholder, varies with the distance between two fireflies. The attractiveness  $\beta$  can be defined as follows [10]:

$$\beta = \beta_0 e^{-\gamma \cdot d^2} \quad (7)$$

where  $d$ ,  $\beta_0$  and  $\gamma$  are the distance of two fireflies, the attractiveness at  $d = 0$ , and  $\gamma$  the light absorption coefficient, respectively.

The distance between two fireflies  $i$  and  $j$  at  $X_i$  and  $X_j$  respectively, is determined as follows:

$$d_{ij} = \|X_i - X_j\| = \sqrt{\sum (x_{i,k} - x_{j,k})^2} \quad (8)$$

where  $x_{i,k}$  is the  $k$ -th parameter of the spatial coordinate  $x_i$  of the  $i$ -th firefly.

In the FA, the movement of a firefly  $i$  towards a more attractive firefly  $X^*$  is determined as follows [10]:

$$X_i^k = X_i^{k-1} + \beta_0 e^{-\gamma d_{ij}^2} (X_j^* - X_i^{k-1}) + \lambda_k (r - 0.5) \quad (9)$$

where the second term is related to the attraction, while the third term is randomization with  $\lambda_r$  being the randomization parameter between 0 and 1;  $r$  is a random number generator uniformly distributed in  $[0, 1]$ .

### 3.2 Standard bat algorithm

BA is a meta-heuristic optimization method developed by Yang [11] based on the echolocation behavior of bats. Bats use echolocation as a hearing based navigation system to detect objects in their surroundings by emitting a sound to the environment. An idealized model of the echolocation can be briefly described as follows: The  $i$ th bat flies at position  $X_i$  with a velocity  $V_i$ , a varying frequency or wavelength and loudness  $A_i$ . It finds prey by tuning its frequency, loudness and pulse emission rate. Furthermore, bats intensify their searches by a local random walk. The fundamental idea behind the BA is that a population including  $n$  bats use echolocation to fly randomly through a  $d$ -dimensional search space updating their positions and velocities.

Each solution  $X_i = \{X_{i1}, X_{i2}, \dots, X_{id}\}^T$  is evaluated by its fitness function value  $f(X_i)$  and the bats' flight aims at finding the best solutions. The positions and velocities of bats have to be updated in search space. The new solutions  $X_i^k$  and velocities  $V_i^k$  at iteration  $k$  are as follows:

$$\Omega_i = \Omega_{\min} + (\Omega_{\max} - \Omega_{\min}) \cdot \beta \quad (10)$$

$$V_i^k = V_i^{k-1} + (X_i^* - X_i^{k-1}) \cdot \Omega_i \quad (11)$$

$$X_i^k = X_i^{k-1} + V_i^k \quad (12)$$

where  $\Omega_{\min}$  and  $\Omega_{\max}$  are the lower and upper bounds imposed for the frequency range of bats;  $\beta \in [0, 1]$  is a vector containing uniformly distribution random numbers;  $X_i^*$  is the current global best solution.

A local search is implemented on a randomly selected bat from the current population as follows:

$$X_{\text{new}}^k = X_{\text{old}}^k + zA^k \quad (13)$$

where  $z \in [-1, +1]$  is a random number;  $A^k$  is the average loudness of all the bats at the current iteration.

As the loudness usually decrease once a bat has found its prey, the rate of pulse emission increases in order to raise the attack accuracy. In this case, the loudness  $A_i$  and the rate  $r_i$  of pulse emission should be updated during the optimization process as follows:

$$A_i^k = \alpha \cdot A_i^{k-1} \quad (14)$$

$$r_i^k = r_i^0 \cdot (1 - e^{-\gamma(k-1)}) \quad (15)$$

where  $\alpha$  (the loudness decay factor) and  $\gamma$  (the pulse increase factor) are constants.

It is clear that  $\alpha$  and  $\gamma$  are two important parameters on the computational performance of BA and their best values can be determined by sensitivity analysis.

### 3.3 Chaotic levy flight bat algorithm

An efficient metaheuristic optimization algorithm should possess balanced exploration and exploitation characteristics. For a metaheuristic with dominant exploration characteristic the convergence rate would be slow while the dominant exploitation characteristic results in trapping in local optima. In the both cases, the metaheuristic is not able to find the global or even near global optimum. The diversification via randomization provides a good way to balance between exploration and exploitation and avoids the solutions being trapped at local optima. Employing a uniform distribution is not the only way to achieve randomization. In fact, random walks such as Levy flights [18] on a global scale are more efficient. Levy flight process is a random walk that is characterized by a series of instantaneous jumps chosen from a probability density function which has a power law tail. This process represents the optimum random search pattern and is frequently found in nature [19]. The Levy flight-based random walk in the  $k$ th step of optimization process is usually represented as follows:

$$X_{new}^k = X_{old}^k + \epsilon \cdot Levy(\lambda) \quad (16)$$

where  $Levy(\lambda)$  is Levy flight and its random steps are drawn from a Levy distribution for large steps as follows:

$$Levy(\lambda) = k^{-\lambda}, \quad 1 \leq \lambda \leq 3 \quad (17)$$

Infinite variance of Levy distribution allows long jumps in design space to regions far from the vicinity of the previous point and this can prevent the optimizer from trapping into local optima. On the other hand, small jumps are required to exploit the optimum solutions in some regions.

With the development of theories and applications of nonlinear dynamics, chaos concept has attracted great attention in various fields [20]. The chaos has the property of the non-repetition, ergodicity, pseudo-randomness and irregularity [21] and the track of chaotic variable can travel ergodically over the whole design space. In the last years, many successful combination of the chaos theory with various metaheuristic algorithms have been reported in literature [22]. The chaos theory and Levy flight were utilized in [20] to improve the performance of BA. The well-known logistic map which exhibits the sensitive dependence on initial conditions was employed to generate the chaotic sequence  $c_2$  for the parameter in Levy flight:

$$c_2(k+1) = 4c_2(k) \cdot (1 - c_2(k)), \quad 0 \leq c_2(0) \leq 1 \quad (18)$$

In the chaotic Levy flight bat algorithm (CLFBA), the following equation was used as a neighbor generation method:

$$X_{new}^k = X_{old}^k + c_2 \cdot Levy(\lambda) \quad (19)$$

Because the chaotic sequence can generate several neighborhoods of suboptimal solutions to maintain the variability in the solutions, it can prevent the search process from

becoming premature. Due to its ergodicity, chaotic sequence can generate several neighborhoods of near-optimal solutions. The algorithm probably converges to a space in the search space where good solutions are denser [20]. In the CLFBA employed in the present work the following frequency updating equation is utilized:

$$\Omega_i = \Omega_{\min} + (\Omega_{\max} - \Omega_{\min}) \cdot c_f \quad (20)$$

The position and velocity of each bat are updated using Eqs (8) and (9), respectively.

#### 4. ADVANCED MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS

In order to solve a MOOP a non-dominated front should be such determined that is close to the true Pareto front while maintaining a good diversity along the resulting Pareto front. In the case of MOOP with many design variables, a suitable MOOA should possess two main features of high convergence rate and maintaining diversity. In the present work, two multi-objective metaheuristics based on FA and BA are presented for multi-objective optimum design of truss structures. The basic concepts of these multi-objective metaheuristics, termed as multi-objective firefly algorithm (MO-FA) and multi-objective chaotic levy-flight bat algorithm (MO-CLFBA), are explained in the following subsections.

##### 4.1 Multi-objective-firefly algorithm

In this study, the FA is extended to produce PF for MOOPs. In the so called MO-FA the search process can be summarized as follows.

1) A number of fireflies are randomly selected from the design space and objective (brightness) values of all the fireflies are evaluated.

2) All non-dominated fireflies are added to the repository. If the repository has no member, all of the non-dominated solutions in the current iteration are included to the repository. Otherwise, all of the non-dominated solutions of current iteration are added to the repository and all dominated members of the repository are removed.

3) The most attractive firefly ( $X_f^k$ ) is updated. In iteration  $k$ , the best solution  $X_f^k$  minimizes the combined objective function  $\psi(X)$  defined as follows:

$$\psi(X) = \sum_{j=1}^n a_j f_j \quad (21)$$

$$a_j = \frac{p_j}{j}, \quad \sum_{j=1}^n a_j = 1 \quad (22)$$

where  $p_j$  are the uniformly distributed random numbers between 0 and 1.

At each iteration the weights  $a_j$  should be chosen randomly, so that the non-dominated solution can sample diversely along the Pareto front [16].

If a firefly is not dominated by others it moves according to Eq. (23). Furthermore, the randomness is reduced as the iterations proceed according to Eq. (24).

$$X_{\text{new}}^k = X_f^k + \lambda_R(r - 0.5) \quad (23)$$

$$A_k = 0.9^k \quad (24)$$

4) A new swarm of fireflies is regenerated for the next iteration in the neighborhood of the non-dominated fireflies of the repository. In [9] a simple equation has been proposed for population regeneration which is utilized in the present work. In this case one new particle  $l$  is generated as follows:

$$X_l = X_r + w(X_{max} - X_{min}) \cdot randn_l \cdot \left(1 - \frac{k}{k_{max}}\right) \quad (25)$$

where  $randn_l$  is a random number from a standard normal distribution which changes for each firefly,  $X_{min}$  and  $X_{max}$  are the minimum and maximum of all design variables in the repository respectively,  $X_r$  is one randomly selected member of the repository, and  $w$  is a parameter which increase the domain of new generated particles and in this study is set to 4.

5) When a termination criterion is meet, the solutions exist in the repository are introduced as the *PF* of the MOOP at hand.

#### 4.2 Multi-objective-chaotic levy flight bat algorithm

In this work, a new and efficient multi-objective-chaotic Levy flight bat algorithm (MO-CLFBA) is proposed to tackle MOOP. In the algorithm, CLFBA acts as optimizer until the loop reaches its maximum limit. For each loop the non-dominated solutions are stored in a repository and all dominated solutions are removed.

In the framework of the proposed MOOA, the described CLFBA in section 3.2 is employed as the optimizer and the search process is accomplished as follows.

1) Individuals of a population of bats are randomly selected from the search space and they are evaluated.

2) All non-dominated bats are added to the repository.

3) The global best position of bats ( $X_g^*$ ) is updated. The global best position is the best solution obtained by neighbors of a bat so far. When solving a single-objective problem, it is completely determined once a neighborhood topology is established. However, in the case of MOOPs, the conflicting nature of multiple objectives makes the choice of a single optimum solution difficult. To resolve this problem, we update the  $X_g^*$  based on the crowding distance (*CD*) [1] value.

$$CD_i = \sum_{j=1}^m \left( \frac{f_{j,i+1} - f_{j,i-1}}{|f_{j,max} - f_{j,min}|} \right) \quad (26)$$

where  $f_j$  is the  $j$ th objective function.

In each iteration the crowding distance value of all elements in the repository is calculated and the higher crowding distance value signifies the best solution. The position  $X_i$  and velocity  $V_i$  of bats are updated. The loudness  $A_i$  and the rate  $r_i$  of pulse emission parameters of bats are updated based on the definition of Pareto dominance. Random walk is performed based on chaos theory and Levy flight described earlier.

4) When a termination criterion is meet, the solutions exist in the repository are introduced as the *PF* of the MOOP at hand.



### 4.3 Handling of design constraints

In the framework of the proposed MOOAs the constraints are handled by a simple method which was utilized in [1,3,9]. In order to compare two bats  $Z^1$  and  $Z^2$ , first they are checked for constraint violation. If both are feasible, then the non-dominance is recognized as winner. If  $Z^1$  is feasible and  $Z^2$  is infeasible,  $Z^1$  dominates. If both are infeasible, then the one with the lowest amount of constraint violation dominates the other one.

## 5. NUMERICAL RESULTS

The computational performance of the proposed MOOAs is compared with that of some other well-known algorithms in the case of constrained structural optimization problems. In order to compare the performance of different MOOAs, PFs obtained by different ones are graphically compared. For all of the presented examples the repository size of 100 and a population of 50 individuals are employed.

### 5.1 10-Bar planar truss

The 10-bar planar truss is illustrated in Figure 1. The objective is to minimize the volume of the structure and the vertical displacement at node 6 simultaneously considering the cross-sectional areas of the 10 truss numbers as design variables. The upper and lower bounds of design variables are 0.1 and 30 in<sup>2</sup>, respectively. External load  $P$  and modulus of elasticity  $E$  are 100000 lb and 10<sup>4</sup> ksi, respectively. Limitations on the axial stress of elements are design constraints and the maximum allowable stress is set to 25 ksi. In this example, maximum number of function evaluations is set to 25,000.

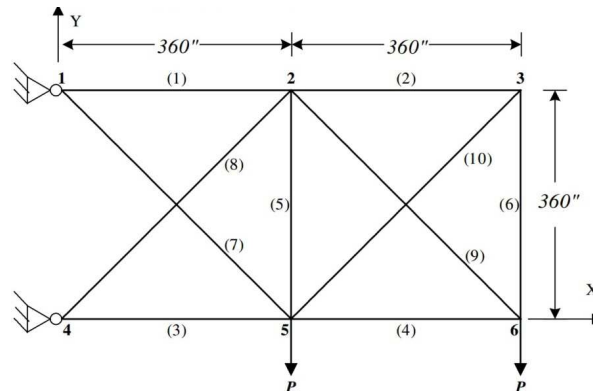


Figure 1. 10-bar planar truss

In this example, the results of MO-FA and MO-CLFBA are compared with those of the constrained multi-objective immune algorithm (C-MOIA) [5]. The obtained PF from different MOOAs are presented in Figure 2. Furthermore, the extreme objective values obtained by the mentioned MOOAs are presented in Table 1.

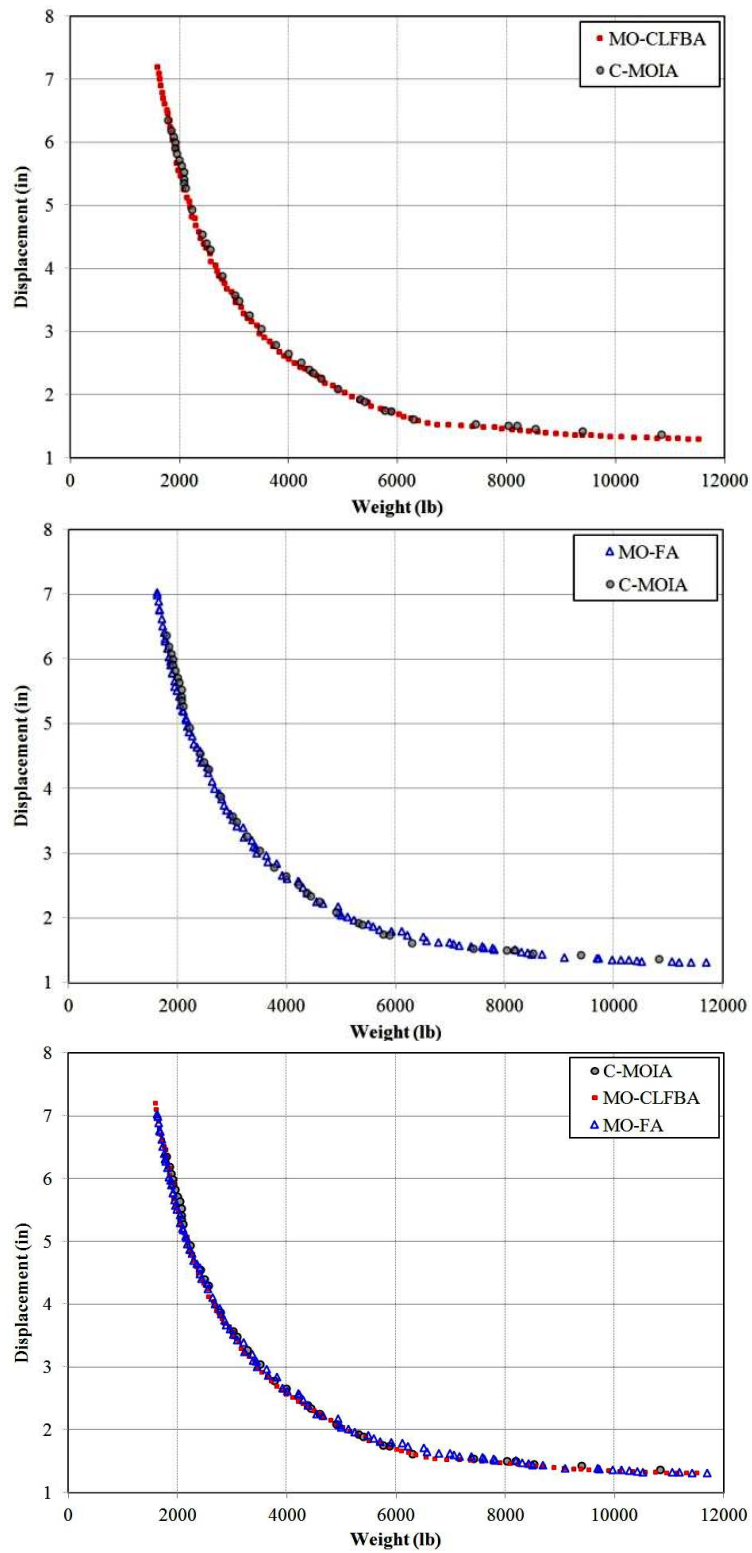


Figure 2. 10-bar planar truss Pareto front for different MOOAs

Table 1: Extreme solutions for 10-bar truss in PF obtained by different MOOAs

MOOA	Extreme point 1 (Minimum Weight)		Extreme point 2 (Minimum Displacement)	
	Weight(Ib)	Displacement (in)	Weight(Ib)	Displacement (in)
C-MOIA [5]	1793.51	6.3562	10841.35	1.3611
MO-CLFBA	1594.34	7.1953	11513.00	1.3034
MO-FA	1625.81	7.0221	11701.20	1.3093

It can be observed that the proposed MO-CLFBA and MO-FA appropriately cover the Pareto front while the other algorithm couldn't achieve this task. Furthermore, the convergence to the Pareto front of the MO-CLFBA and its ability in covering all parts of it is better in comparison with the MO-FA.

### 5.2 25-Bar space truss

Figure 3 shows the 25-bar space truss structure. Members of the structure are divided into eight groups, as follows: (1)  $A_1$ , (2)  $A_2$ – $A_5$ , (3)  $A_6$ – $A_9$ , (4)  $A_{10}$ – $A_{11}$ , (5)  $A_{12}$ – $A_{13}$ , (6)  $A_{14}$ – $A_{17}$ , (7)  $A_{18}$ – $A_{21}$  and (8)  $A_{22}$ – $A_{25}$ . The loading details are as:  $F_{x(1)}=4.45$  kN,  $F_{y(1)}=-44.5$  kN,  $F_{z(1)}=-44.5$  kN,  $F_{y(2)}=-44.5$  kN,  $F_{z(2)}=-44.5$  kN,  $F_{x(3)}=2.25$  kN and  $F_{x(6)}=2.67$  kN.

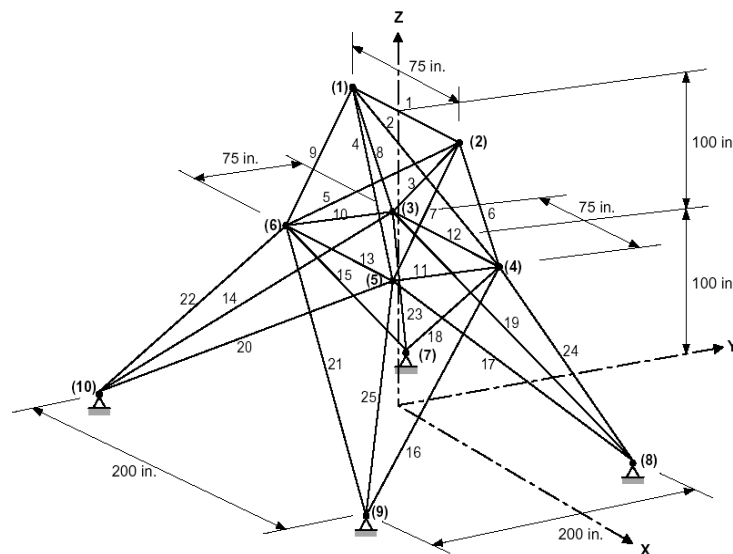


Figure 3. 25-bar space truss

In this example, the objective functions to be minimized are the structural weight and the displacement in Y-direction at node 1. The upper and lower bounds for the cross sections of truss elements are 0.1 and 3.4 in<sup>2</sup>, respectively. Modulus of elasticity and weight density are  $E=10^4$  ksi and  $\rho=0.11\text{lb/in}^3$ , respectively. As the design constraints the axial stress in truss elements is limited to 40 ksi. In the present example maximum number of function evaluations is set to 30,000. The results of proposed algorithms in this study are compared

with those of the charge system search multi-objective particle swarm optimization (CSS-MOPSO) [7]. The PF obtained by different MOOAs are presented in Figure 4.

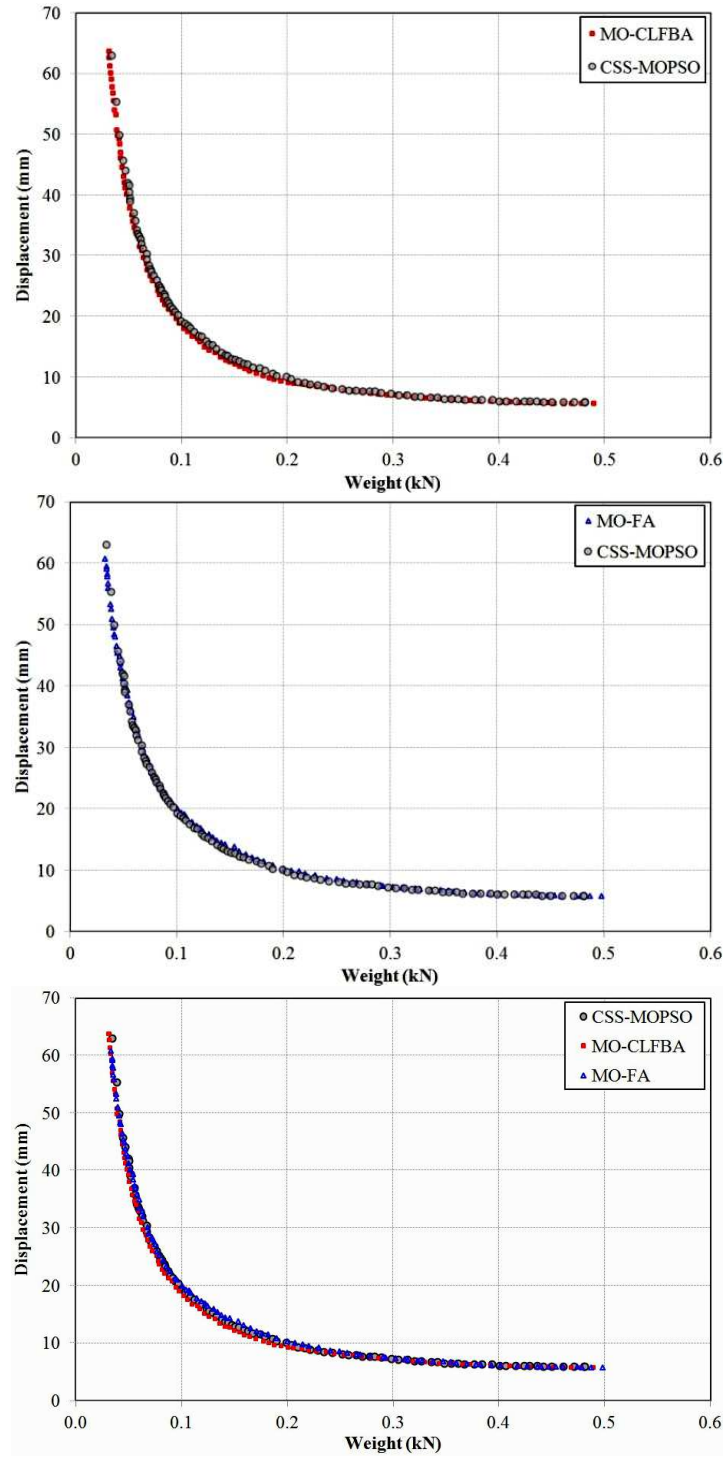


Figure 4. 25-bar space truss Pareto front for different MOOAs

Table 2 compares the extreme objective values obtained by MO-CLFBA and MO-FA with those of the CSS-MOPSO [7]. The results imply that the proposed MO-CLFBA outperforms the CSS-MOPSO [7] and MO-FA in terms of the convergence to the Pareto front and ability in covering all parts of it.

Table 2: Extreme solutions for 25-bar space truss in PF obtained by different MOOAs

MOOA	Extreme point 1 (Minimum Weight)		Extreme point 2 (Minimum Displacement)	
	Weight(kN)	Displacement (mm)	Weight(kN)	Displacement (mm)
CSS-MOPSO [7]	4.8111	5.8437	0.3440	62.9807
MO-CLFBA	4.8916	5.8108	0.3125	63.7685
MO-FA	4.9796	5.8093	0.3270	60.7876

### 5.3 56-Bar space dome

The third example of the present study is a 56-bar space dome illustrated in Figure 5.

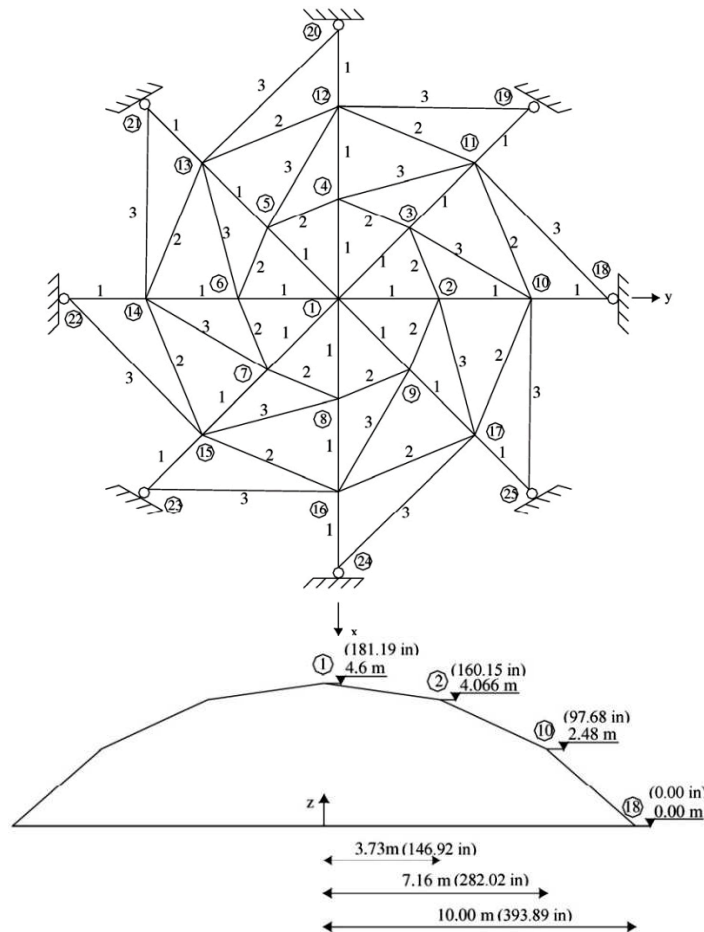


Figure 5. 56-bar space dome

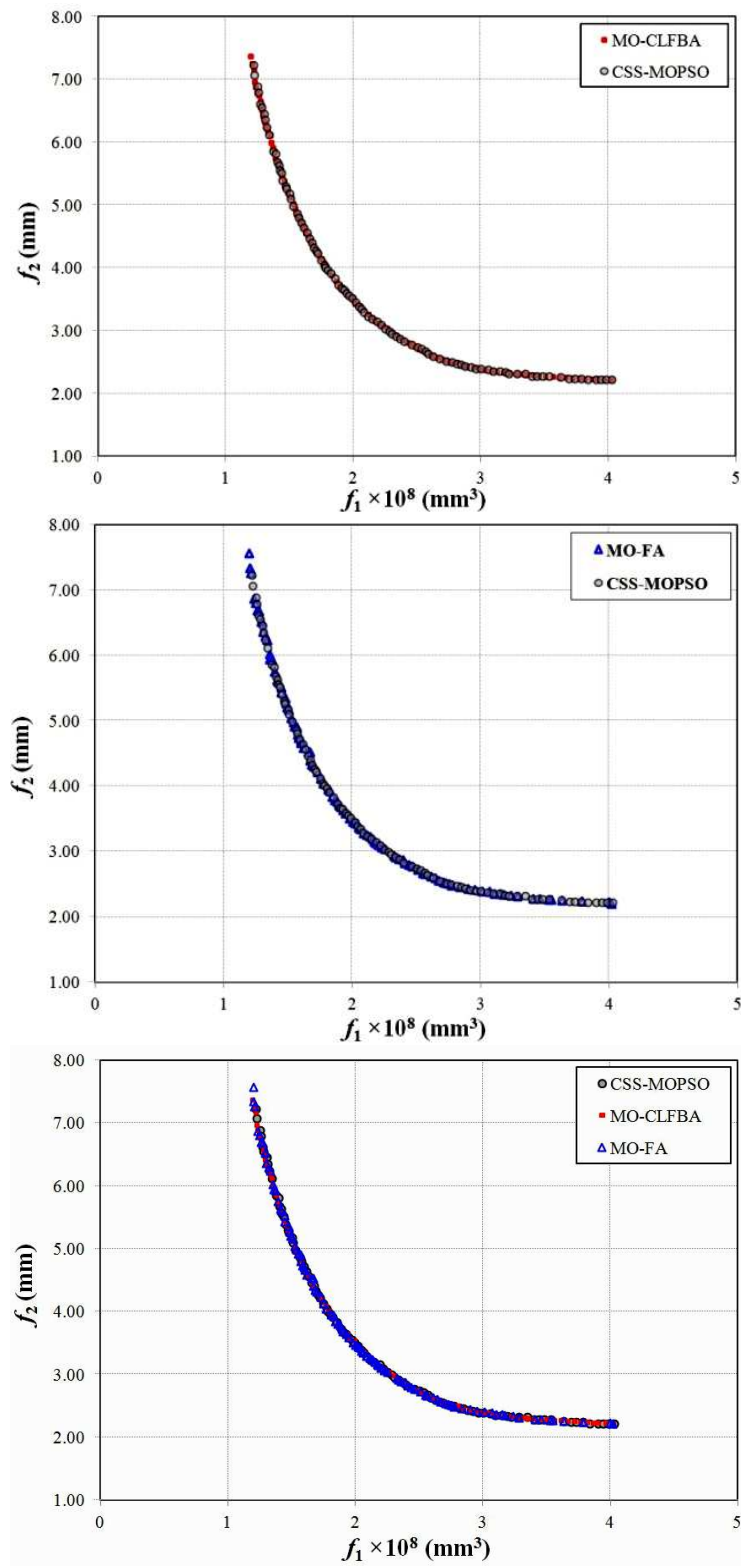


Figure 6. 56-bar space dome Pareto front for different MOOAs

All free nodes are loaded with 4 kN in the Y-direction. In the Z-direction Joint 1 is loaded with 30 kN, while the remaining free nodes are loaded with 10 kN. The modulus of elasticity is set to 210 kN/mm<sup>2</sup> and the lower and upper bounds on the member cross-sectional areas are taken as 200 and 2000 mm<sup>2</sup>, respectively. The vertical displacements of joints 4, 5, 6, 12, 13 and 14 is restricted to 4 mm and the displacement of joint 8 in the Y-direction is limited to 2 mm.

The objective functions which should be minimized in this example are as follows:

$$f_1(x) = \sum_{i=1}^n A_i l_i \quad (27)$$

$$f_2(x) = \sqrt{\delta_{1x}^2 + \delta_{1y}^2 + \delta_{1z}^2} \quad (28)$$

where  $A_i$  and  $l_i$  are the cross-sectional area and length of the  $i$ th member;  $\delta_{1x}$ ,  $\delta_{1y}$  and  $\delta_{1z}$  are displacements of Joint 1 in X-, Y- and Z- directions, respectively.

In the present example the number of function evaluations is limited to 30,000. The results of MO-CLFBA and MO-FA are compared with those of the CSS-MOPSO [7]. Figure 6 compares the PF of the mentioned MOOAs for this example. The extreme objective values obtained by MO-CLFBA and MO-FA are compared with those of the CSS-MOPSO [7] in Table 3.

Table 3: Extreme solutions for 56-bar space dome in PF obtained by different MOOAs

MOOA	Extreme point 1 (Minimum Weight)		Extreme point 2 (Minimum Displacement)	
	$f_1 \times 10^8$ (mm <sup>3</sup> )	Displacement (mm)	$f_1 \times 10^8$ (mm <sup>3</sup> )	Displacement (mm)
CSS-MOPSO [7]	4.02923368	2.2148	1.20812690	7.5495
MO-CLFBA	4.02417574	2.2138	1.19803356	7.3713
MO-FA	4.02111310	2.2095	1.19926203	7.5629

The obtained numerical results indicate that the proposed MO-CLFBA and MO-FA are slightly better than the CSS-MOPSO [7] in terms of the convergence to the Pareto front and covering all parts of it. In addition the computational performance of the MO-CLFBA and MO-FA are almost the same.

## 6. CONCLUSIONS

In the present work, two advanced MOOAs are proposed for tackling multi-objective optimization problems of truss structures. The main optimization engines in the presented MOOAs are FA and CLFBA and thus these algorithms are termed as MO-FA and MO-CLFBA, respectively. In the frameworks of MO-FA a new operator is employed for regeneration of fireflies in the neighboring regions of the non-dominated fireflies of the repository. This operator improves the overall computational performance of the algorithm. For the MO-CLFBA chaotic sequence and Levy flight are combined to efficiently achieve

the optimization task. The numerical results demonstrate that the proposed MOOAs possess appropriate computational abilities in solving complex multi-objective optimization problems in comparison with other existent algorithms.

## REFERENCES

1. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi objective genetic algorithm: NSGA-II, *IEEE Trans Evol Comput* 2002; **6**: 182–97.
2. Knowles JD, Corne DW. Approximating the nondominated front using the Pareto archived evolution strategy, *Evol Comput* 2000; **8**: 149–72.
3. Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization, *IEEE Trans Evolution Comput* 2004; **8**: 256–79.
4. Coello CA, Christiansen AD. Multiobjective optimization of trusses using genetic algorithms, *Comput Struct* 2000; **75**: 647–60.
5. Luh GC, Chueh CH. Multi-objective optimal design of truss structure with immune algorithm, *Comput Struct* 2004; **82**: 829–44.
6. Kaveh A, Kalateh-Ahani M, Fahimi-Farzam M. Constructability optimal design of reinforced concrete retaining walls using a multi-objective genetic algorithm, *Struct Eng Mech* 2013; **47**: 227–45.
7. Kaveh A, Laklejadi K. A hybrid multi-objective optimization and decision making procedure for optimal design of truss structures, *IJST Trans of Civil Eng* 2011; **35**: 137–54.
8. Richardson JN, Adriaenssens S, Bouillard P, Coelho RF. Multiobjective topology optimization of truss structures with kinematic stability repair, *Struct Multidisc Optim* 2012; **46**: 513–32.
9. Kaveh A, Laknejadi K. A new multi-swarm multi-objective optimization method for structural design, *Adv Eng Softw* 2013; **58**: 54–69.
10. Yang XS. Firefly algorithms for multimodal optimization, In: Stochastic Algorithms: Foundations and Applications (Eds O. Watanabe and T. Zeugmann), SAGA 2009, Lecture Notes in Computer Science, 5792, Springer-Verlag, Berlin, 169–178, 2009.
11. Yang XS. A new metaheuristic bat-inspired algorithm. In: Gonzalez JR et al. (eds) Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). Studies in Computational Intelligence, vol284. Springer: Berlin; 65–74, 2010.
12. Coello CAC, Van Veldhuizen DA, Lamont GB. *Evolutionary Algorithms for Solving Multi-Objective Problems*, New York, USA: Kluwer Academic Publishers; 2002.
13. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer Verlag; 2014.
14. Gholizadeh S, Barati H. Topology optimization of nonlinear single layer domes by a new metaheuristic, *Steel Compos Struct* 2014; **16**: 681–701.
15. Gholizadeh S, Shahrezaei AM. Optimal placement of steel plate shear walls for steel frames by bat algorithm, *Struct Des Tall Spec Buil* 2014. (in press)
16. Yang XS. Multiobjective firefly algorithm for continuous optimization, *Eng Comput* 2013; **29**: 175–84.



17. Yang XS. Bat Algorithm for Multiobjective Optimization, *Int J Bio-Inspired Comput* 2011; **3**: 267–74.
18. Pavlyukevich I. Lévy flights, non-local search and simulated annealing, *J Comput Phys* 2007; **226**: 1830–44.
19. Viswanathan GM. Lévy flights and superdiffusion in the context of biological encounters and random searches, *Phy Life Rev* 2008; **5**: 133–50.
20. Lin JH, Chou CW, Yang CH, Tsai HL. A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. *J Comput Inf Tech* 2012; **2**: 56–63.
21. Pecora L, Carroll T. Synchronization in chaotic system, *Phys Rev Lett* 1990; **64**: 821–24.
22. Gandomi AH, Yang XS. Chaotic bat algorithm, *J Comput Sci* 2014; **5**: 224–32.