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

A Comparative Study of Three Metaheuristics for Optimum Design of Engineering Structures

Serdar Carbas[1], Ibrahim Aydogdu[2], Mehmet Polat Saka[3]

[1] Middle East Technical University, Department of Engineering Sciences, Ankara, Turkey, carbas@metu.edu.tr
[2] Akdeniz University, Department of Civil Engineering, Antalya, Turkey, aydogdu@akdeniz.edu.tr
[3] University of Bahrain, Department of Civil Engineering and Architecture, Isa Town, Bahrain, mpsaka@eng.uob.bh

1. Abstract
A comparative study is carried out on the optimum design of a real-size steel frame by considering three different metaheuristic search techniques. The techniques are selected as the Firefly Algorithm (FFA), Artificial Bee Colony (ABC), and Cuckoo Search (CS) algorithms. Metaheuristic search techniques of optimization are non-deterministic methods and they rely on heuristics in finding the better solutions in the search space. They use random or probabilistic parameters, while they search for the optimum solution, rather than deterministic quantities. The source of random variables may be several depending on the nature and the type of problem. The heuristics behind these innovative techniques is borrowed from the nature or physics. In the design example considered, the design constraints include the displacement limitations, inter-story drift restrictions, strength requirements for beams and beam-columns which are formulated according to provisions of LRFD-AISC (Load and Resistance Factor Design of American Institute of Steel Institution).
2. Keywords: Optimum Design of Engineering Structures; Load and Resistance Factor Design; Combinatorial Optimization; Metaheuristic Search Techniques.

3. Introduction

The saving that can be achieved in the material cost of structures is one of the major factors in the construction industry due to increasing human population and diminishing resources in today's world. The optimum design methods used for minimizing the weight of structures intend to achieve the optimum design by finding out the optimum values for set of design variables under certain design criteria. The optimum design of steel structures is generally achieved by designating steel sections from an available list to the members of steel frames such that the objective function is minimized while all of the design constraints are satisfied [1]. Since steel frame structures are widely used for structural applications, optimum design of this type of structures has a great importance. Generally, in design optimization of frame structures, the objective is to find the best feasible structure with the minimum weight [2]. As a results of developments took place in metaheuristic techniques in last three decades, large number of metaheuristic approaches have been developed which are employed in the evolvement of several structural optimization methods. The main idea behind metaheuristic algorithms is to use some heuristic taken from natural phenomena, physics, social sciences and swarm intelligence. The latest application of these techniques has shown that they are quite successful in finding the optimum solution of real world problems regardless of their complexity compare to classical optimization methods. The practical advantage of metaheuristics lies in both their effectiveness and general applicability. In fact, metaheuristics are the most general kinds of stochastic optimization algorithms, and are applied to a very wide range of engineering design problems [3]. Firefly algorithm (FFA), artificial bee colony (ABC), and cuckoo search algorithm are among the most recent metaheuristics developed metaheuristic algorithms. The FFA is a novel optimization technique originated by Yang [4]. It is inspired by social behavior of fireflies and the phenomenon of the use of bioluminescent in communication. Despite being a recently developed method by Karaboga and Basturk [5-8], the ABC algorithm is applied to many design optimization problems. This optimization technique mimics intelligent behavior of honey bee swarm. CS algorithm simulates the breeding characteristic of certain cuckoo species into a numerical optimization technique [9]. In the present study, FFA, ABC and CS algorithms are employed to achieve the minimum weight design of a steel frame structure and their performance in obtaining the optimum solutions is compared.

4. Optimum Design of Space Steel Frames

Optimum design of a space steel frame necessitates the selection of steel profiles for its members from steel section list available in standards such that serviceability and strength limitations specified by the code of practice are satisfied and its weight is the minimum. Therefore, the objective function of optimum design of space steel frame problems is taken as the minimum weight of the frame as given in Eq. 1. In this equation, W is the weight of the frame, x is the vector of steel sections to be assigned for each group in the space frame which are taken as design variables. x represents the sequence number of the steel section that is adopted to each member group in the steel space frame. m_r is the unit weight of the steel section selected from the standard steel sections table that is adopted for group r, t_r is the total number of members in group r and NG is the total number of groups in the frame, l_s is the length of member which belongs to group r.

$$\text{Minimize, } W(x) = \sum_{r=1}^{NG} m_r \cdot \sum_{s=1}^{t_r} l_s \tag{1}$$

Strength constraint function g_s is defined from inequalities given in Chapter H of LRFD-AISC[10] as:

$$g_s\ x = \frac{P_u}{\varnothing P_n} + \frac{8}{9}\left(\frac{M_{ux}}{\varnothing_b M_{nx}} + \frac{M_{uy}}{\varnothing_b M_{ny}}\right) - 1.0 \leq 0 \quad for\ \frac{P_u}{\varnothing P_n} \geq 0.2 \tag{2}$$

$$g_s\ x = \frac{P_u}{2\varnothing P_n} + \left(\frac{M_{ux}}{\varnothing_b M_{nx}} + \frac{M_{uy}}{\varnothing_b M_{ny}}\right) - 1.0 \leq 0 \quad for\ \frac{P_u}{\varnothing P_n} < 0.2 \tag{3}$$

where, M_{nx} is the nominal flexural strength at strong axis (x axis), M_{ny} is the nominal flexural strength at weak axis (y axis), M_{ux} is the required flexural strength at strong axis (x axis), M_{uy} is the required flexural strength at weak axis (y axis), P_n is the nominal axial strength (Tension or compression) and P_u is the required axial strength (Tension or compression) for member i. The values of M_{ux} and M_{uy} are to be obtained by carrying out P-Δ analysis of the steel frame. This is an iterative process which quite time consuming. In Chapter C of LRFD-AISC an alternative way is suggested for the computations of M_{uy} and M_{ux} values. In this way two first order elastic analyses are carried out. In the first, frame is analyzed under the gravity loads only where the sway of the frame is prevented to obtain M_{nt} values. In the second, the frame is analyzed only under the lateral loads to find M_{lt} values. These moment values are combined as given in the following.

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

where, B_1 is the moment magnifier coefficient and B_2 is the sway moment magnifier coefficient. The details of how these coefficients are calculated are given in Chapter C of LRFD-AISC [10].
Deflection g_d, top story drift g_{td}, and inter story drift g_{id} constraint functions are given in the following Eqs (5), (6) and (7) respectively [11].

$$g_d(x) = \frac{\delta_{jl}}{L/Ratio} - 1.0 \leq 0 \quad j = 1, 2, \ldots, n_{sm} \quad l = 1, 2, \ldots, n_{lc} \tag{5}$$

$$g_{td} = \frac{\Delta_{jl}^{top}}{H/Ratio} - 1.0 \leq 0 \quad j = 1, 2, \ldots, n_{jtop} \quad l = 1, 2, \ldots, n_{lc} \tag{6}$$

$$g_{id} = \frac{\Delta_{jl}^{oh}}{h_{sx}/Ratio} - 1.0 \leq 0 \quad j = 1, 2, \ldots, n_{st} \quad l = 1, 2, \ldots, n_{lc} \tag{7}$$

where, δ_{jl} is the maximum deflection of j^{th} member under the l^{th} load case, L is the length of member, n_{sm} is the total number of members where deflections limitations are to be imposed, n_{lc} is the number of load cases, H is the height of the frame, n_{top} is the number of joints on the top story, $\Delta^{top}{}_{jl}$ is the top story displacement of the j^{th} joint under l^{th} load case, n_{st} is the number of story, n_{lc} is the number of load cases and $\Delta^{oh}{}_{jl}$ is the story drift of the j^{th} story under l^{th} load case, h_{sx} is the story height and *Ratio* is limitation ratio for lateral displacements described in ASCE Ad Hoc Committee report [12] . According to the ASCE Ad Hoc Committee report, the accepted range of drift limits by first-order analysis is *1/750* to *1/250* times the building height H with a recommended value of *H/400*.The typical limits on the inter-story drift are *1/500* to *1/200* times the story height. Based on this report the deflection limits recommended are proposed for general use which is repeated in Table 1.
Geometric limitations g_{cc}, and g_{bc} as shown in Figure 1 are included in the design problem as in the Eqs (8) and (9); where n_{ccj} is the number of column-to-column geometric constraints defined in the problem , $m^a{}_i$ is the unit weight of W section selected for above story, $m^b{}_i$ is the unit weight of W section selected for below story, $D^a{}_i$ is the depth of W section selected for above story, $D^b{}_i$ is the depth of W section selected for below story, n_{j1} is the number of joints where beams are connected to the web of a column, n_{j2} is the number of joints where beams connected to the flange of a column, D^{ci} is the depth of W section selected for the column at joint i, $t^{ci}{}_b$ is the flange thickness of W

section selected for the column at joint *i*, B^{ci}_f is the flange width of *W* section selected for the column at joint *i* and B^{bi}_f is the flange width of *W* section selected for the beam at joint *i*.

$$g_{cc}(x) = \sum_{i=1}^{n_{ccj}} \left(\frac{D_i^a}{D_i^b} - 1.0 \right) + \sum_{i=1}^{n_{ccj}} \left(\frac{m_i^a}{m_i^b} - 1.0 \right) \leq 0 \tag{8}$$

$$g_{bc}(x) = \sum_{i=1}^{n_{js}} \left(\frac{B_i^{bi}}{D^{ci} - 2t_b^{ci}} - 1.0 \right) \leq 0 \text{ or } \sum_{i=1}^{n_{j2}} \left(\frac{B_f^{bi}}{B_f^{ci}} - 1.0 \right) \leq 0 \tag{9}$$

Table 1: Displacement limitations for steel frames.

	Item	Deflection Limit
1	Floor girder deflection for service live load	L/360
2	Roof girder deflection	L/240
3	Lateral drift for service wind load	H/400
4	Inter story drift for service wind load	H/300

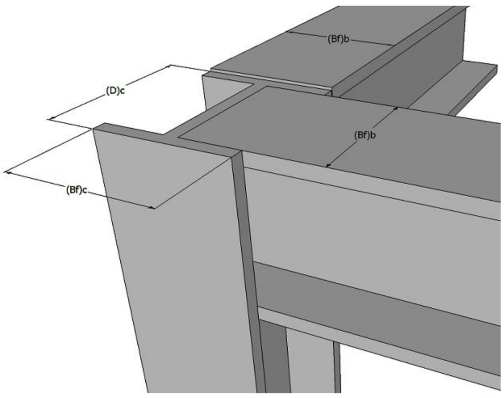


Figure 1: Beam column geometric constraints

5. Metaheuristic Techniques

Stochastic optimization is the general class of techniques which employ some degree of randomness to find optimal solutions to difficult problems. In order to comprehensively explore a wide design space, stochastic search techniques reveal their promising abilities in comparison with gradient-based optimization methods. Metaheuristics are the most general of these kinds of algorithms, and are applied to a very wide range of problems. Metaheuristics solve instances of problems that are believed to be hard in general, by exploring the usually large solution search space of these instances. These algorithms achieve result by reducing the effective size of the space and by exploring that space efficiently. Metaheuristics serve three main purposes: solving problems faster, solving large problems, and obtaining robust algorithms. Moreover, they are simple to design and implement, and are very flexible. Most of the metaheuristics mimic natural metaphors to solve complex optimization problems such as evolution of species, annealing process, ant colony, particle swarm, immune system, bee colony, and wasp swarm. Metaheuristics are more and more popular in different research areas [13-16].

For all metaheuristic methods discussed in the present study, a penalty integrated objective function is defined to evaluate infeasible designs in proportion to the sum of the constraint violation.

$$W_p = W.(1+C)^{\varepsilon} \tag{10}$$

where, *W* is the value of objective function given in Eq. (1), W_p is the penalized weight of frame, *C* is the total constraint violation values calculated from the sum of the constraints violation functions shown in Eq. (11), ε is penalty coefficient taken as 2.0.

$$C = \sum_{i=1}^{NC} g_i(x) = \Sigma g_s\ x + \Sigma g_d\ x + \Sigma g_{td}\ x + \Sigma g_{id}\ x + \Sigma g_{cc}\ x + \Sigma g_{bc}(x) \geq 0 \quad (11)$$

where, *C* are the constraints violation functions for strength, deflection, drift and geometric constraints given in inequalities defined in Eqs. (2) to (9). In general form, constraints violation functions can be expressed as:

$$C_i = \begin{cases} 0 & \text{if } g_i(x_j) \leq 0 \quad i=1,\ldots NC \\ g_i(x_j) & \text{if } g_i(x_j) \leq 0 \quad j=1,\ldots NG \end{cases} \quad (12)$$

where, $g_i(x)$ is i^{th} constraints function, x is the vector of design variables, *NC* is the number of constraint functions and *NG* is the total number of member groups in the optimization problem.

5.1 Adaptive Firefly Algorithm (AFFA)

The Firefly Algorithm (FFA) is a very recent heuristic optimization algorithm developed by Yang [4] and is inspired by the flashing behavior of fireflies. According to Yang [4], FFA algorithm has three basic rules; (i) All fireflies are unisex, so that one firefly is attracted to other fireflies regardless of their sex, (ii) Attractiveness is proportional to brightness, so for any two flashing fireflies, the less bright firefly will move towards the brighter firefly. Both attractiveness and brightness decrease as the distance between fireflies increases. If there is no firefly brighter than a particular firefly, that firefly will move randomly, (iii) The brightness of a firefly is affected or determined by the landscape of the objective function. There are two essential components to FFA; the variation of light intensity and the formulation of attractiveness. The latter is assumed to be determined by the brightness of the firefly, which in turn is related to the objective function of the problem under study. As the light intensity and attractiveness decrease and the distance from the source increases, the variation of light intensity and attractiveness should be a monotonically decreasing function. For example, the light intensity can be expressed as

$$I(r_{ij}) = I_0 e^{-\gamma r_{ij}^2} \quad (13)$$

where the light absorption coefficient γ is a parameter of the FFA and r_{ij}, is the distance between fireflies *i* and *j* at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively, which can be defined as the Cartesian distance $r_{ij} = \left\| x_i - x_j \right\|$. Because a firefly's attractiveness is proportional to the light intensity seen by other fireflies, it can be defined by

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (14)$$

where β_0 is the attractiveness at $r = 0$. Finally, the probability of a firefly *i* being attracted to another, more attractive (brighter) firefly *j* is determined by

$$\Delta x_i = \beta_0 e^{-\gamma r_{ij}^2}\ x_j^t - x_i^t\ + \alpha\varepsilon_i\,,\ x_i^{t+1} = x_i^t + \Delta x_i \quad (15)$$

where *t* is the generation number, ε_i is a random vector (*e.g.*, the standard Gaussian random vector in which the mean is 0 and the standard deviation is 1) and α is the randomization parameter. The first term on the right-hand side of Eq. (15) represents the attraction between the fireflies and the second term is the random movement. In other words, Eq. (15) shows that a firefly will be attracted to brighter or more attractive fireflies and also move randomly. Eq. (15) indicates that the user must set parameters β_0, γ, and α and the distribution of ε_i before initiating the application of FFA, and also shows that there are two limit cases when γ is small or large. a) If γ approaches zero, the attractive and brightness are constants, and consequently, a firefly can be seen by all other fireflies. In this case, the FFA reverts to the PSO. b) If γ approaches infinity, the attractiveness and brightness approach zero, and all fireflies are short-sighted or fly in a foggy environment, moving randomly. In this case, the FFA reverts to the pure random search algorithm. Hence, the FFA generally corresponds to the situation falling between these two limit cases.

5.1.1. Adaptive randomness parameter stage

In classical firefly algorithm, randomness parameter α is taken as static. Selecting randomness parameter as lower value, stagnation and local convergence can be seen in the large scale optimization problems. On the other hand, selecting randomness parameter as higher value, convergence problems might occur in the solution process of optimization problems. In order to overcome these discrepancies, adaptive randomness parameter strategy is suggested. In this strategy, randomness parameter (α) changes dynamically as expressed in the following equation.

$$\alpha^{i} = \alpha_{\max} - (\alpha_{\max} - \alpha_{\min}) \cdot \left(\frac{I^{i}_{\max} - I^{i}_{mean}}{I^{i}_{\max} - I^{i}_{\min}} \right) \tag{16}$$

Equation (16) is adopted from Coello [17]. In this equation α^i represents randomness parameters at cycle *i*. α_{max} and α_{min} represent maximum and minimum randomness parameters defined in the algorithm respectively. I^i_{max} and I^i_{min} represent maximum light density, minimum light density and mean value of light densities of all fireflies at cycle *i* respectively.

5.2 Artificial Bee Colony (ABC)

The artificial bee colony (ABC) algorithm, one of the recent metaheuristic search algorithms, was originated by Karaboga and Basturk [5-8]. This algorithm is based on characteristic behavior of honey bee swarms. In the artificial bee colony algorithm, all bees are categorized in three main groups. These are employed bees, onlooker bees and scout bees. The first group of bees is the *employed bees* that locate food source, evaluate its amount of nectar and keep the location of better sources in their memory. These bees when fly back to hive they share this information to other bees in the dancing area by dancing. The dancing time represents the amount of nectar in the food source. The second group is the *onlooker bees* who observe the dance may decide to fly to the food source if they find it is worthwhile to visit the food source. Therefore food sources that are rich in the amount of nectar attract more onlooker bees. The third group is *scout bees* that explore new food sources in the vicinity of the hive randomly. The employed bee whose food source has been abandoned becomes a scout bee. Each employed bee in the colony goes to one food source and this food source is selected only by one employed bee. Therefore, number of employed bees in the artificial bee colony algorithm is equal to number of food sources. By the time the food source is exhausted, onlooker and employed bees of this food source are replaced by scout bees. Then, these bees start finding new food sources through random search. For structural design problems, all randomly selected designs are represented as food sources which are used by bee swarm in the artificial bee colony algorithm. Amount of nectar in each food source represents the weight of the structure. Steps of artificial bee colony algorithm on which structural optimization algorithm is based is as follows:

Step1: Search parameters of artificial bee colony algorithm are initialized in this step. These are number of employed bees (*NEB*), number of onlooker bees (*NOB*), number of cycles and control parameter adjusting the food source (limit). In the algorithm, number of onlooker bees is decided to be equal to number of employed bees.

Step2: After defining search parameters, all foragers in the colony search food source randomly. This means *NEB*+*NOB* frame designs are generated randomly.

Step3: Bees having the best frame designs become employed bees. Then, employed bees start to generate a new frame designs by using the old one as follows:

$$v_{ij} = x_{ij} + \phi_{ij} \cdot (x_{ij} - x_{kj}),\ i,k = 1,2,\cdots,NEB,\ j = 1,2,\cdots,NG \tag{17}$$

where, *i* represents employed bee number index, *k* and *j* are randomly chosen indexes. Although *k* is generated randomly, it is not equal to *k*. $Ø_{ij}$ is a uniformly distributed random number between [-1, 1]. This parameter adjusts size of neighborhood frame design region. Then, new frame designs generated from employed bees are evaluated and their penalized weights are calculated by using aforementioned process. After evaluation process, penalized weights of new frame designs and old frame designs are compared. If penalized weight of the new frame design is better than the old one, the old frame design is replaced with the new one. This process is called greedy selection.

Step 4: After finding new frame designs and replacements, all employed bees return their hive and start their waggle dance. Waggle dance of employed bees are related to penalized weight of frame designs. The remainders of the bees (onlooker bees) watch the waggle dance and make a decision. This decision process of each onlooker bee depends on its probability value associated with frame design. Probability of i^{th} frame design is calculated according to i^{th} onlooker bee by using following function

$$P_i = \frac{(W_p)_i}{\sum_{i=1}^{NOB} (W_p)_i} \tag{18}$$

Then onlooker bees generate new frame designs by using formula in Eq. (18) and make greedy selection the same

as the case of employed bees.
Step 5: If frame design cannot be replaced with the old frame design, this frame design is abandoned and the employed bee associated with that fame design becomes a scout bee. Scout bees generate new frame designs by using random selection process the same as step 2.
Step 6: The steps 3 and 5 are repeated until a pre-assigned maximum iteration number is reached.

5.3 Cuckoo Search (CS)

Cuckoo search (CS) algorithm is originated by Yang and Deb [9] which simulates reproduction strategy of some cuckoo bird species. These species of cuckoo birds lay their eggs in the nests of other birds so that when the eggs are hatched their chicks are fed by the other birds. Sometimes they even remove existing eggs of host nest in order to give more probability of hatching of their own eggs. Some species of cuckoo birds are even specialized to mimic the pattern and color of the eggs of host birds so that host bird could not recognize their eggs which give more possibility of hatching. In spite of all these efforts to conceal their eggs from the attention of host birds, there is a possibility that host bird may discover that the eggs are not its own. In such cases the host bird either throws these alien eggs away or simply abandons its nest and builds a new nest somewhere else. In cuckoo search algorithm cuckoo egg represents a potential solution to the design problem which has a fitness value. The algorithm uses three idealized rules as given in [9]. These are: a) each cuckoo lays one egg at a time and dumps it in a randomly selected nest. b) the best nest with high quality eggs will be carried over to the next generation. c) the number of available host nests is fixed and a host bird can discover an alien egg with a probability of $P_a \in [0,1]$. In this case the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location. Cuckoo search algorithm initially requires selection of a population of n eggs each of which represents a potential solution to the design problem under consideration. This means that it is necessary to generate n solution vector of $x = x_1, \cdots, x_{ng}{}^T$ in a design problem with *ng* variables. For each potential solution vector the value of objective function $f(\boldsymbol{x})$ is also calculated. The algorithm then generates a new solution $x_i^{v+1} = x_i^v + \beta\lambda$ for cuckoo *i* where x_i^{v+1} and x_i^v are the previous and new solution vectors. β >1 is the step size which is selected according to the design problem under consideration. $\boldsymbol{\lambda}$ is the length of step size which is determined according to random walk with Levy flights. A random walk is a stochastic process in which particles or waves travel along random trajectories consists of taking successive random steps. The search path of a foraging animal can be modeled as random walk. A Levy flight is a random walk in which the steps are defined in terms of the step-lengths which have a certain probability distribution, with the directions of the steps being isotropic and random. Hence Levy flights necessitate selection of a random direction and generation of steps under chosen Levy distribution.
Mantegna [18] algorithm is one of the fast and accurate algorithm which generates a stochastic variable whose probability density is close to Levy stable distribution characterized by arbitrary chosen control parameter α ($0.3 \le \alpha \le 1.99$) . Using the Mantegna algorithm, the step size λ is calculated as

$$\lambda = \frac{x}{|y|^{1/\alpha}} \tag{19}$$

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

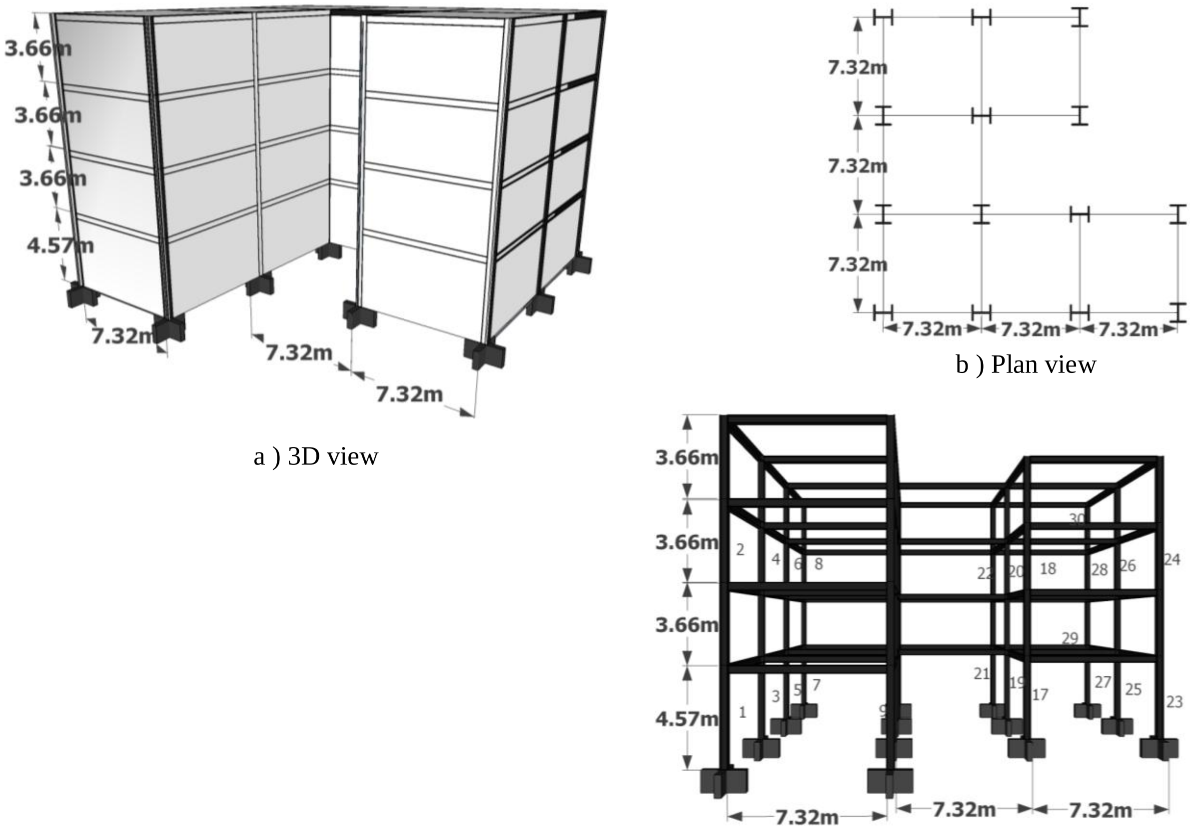
$$\sigma_x(\alpha) = \left[\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\Gamma((1+\alpha)/2)\alpha 2^{(\alpha-1)/2}} \right]^{1/\alpha} \text{ and } \sigma_y(\alpha) = 1 \text{ for } \alpha = 1.5 \tag{20}$$

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

6. Design Example

132-members steel space frame is selected as a design example for comparison of the performance of three metaheuristic algorithms. This three dimensional irregular space steel frame is previously designed in [19-21] by using various metaheuristic algorithms. 3-D, plan and side views of this frame are shown in Figure 2. The space frame consists of 70 joints and 132 members that are grouped into 30 independent design groups. The frame is subjected to gravity loads and lateral loads, which are computed according to ASCE 7-05 [22] based on the following design values: a design dead load of 2.88kN/m^2, a design live load of 2.39kN/m^2, a ground snow load of 0.755kN/m^2. The basic wind speed is considered as 85mph (38 m/s) [22]. The factored distributed gravity loads on the beams of the roof and floors are tabulated in Table 2 and the unfactored earthquake loads are given in Table 3.

The load and combination factors are applied according to code specification [10, 23] as: Load Case1: 1.4D, Load Case2: 1.2D+1.6L+0.5S; Load Case3: 1.2D+0.5L+1.6S; Load Case4: 1.2D+1.0EX+0.5L+0.2S; Load Case5: 1.2D+1.0EZ+0.5L+0.2S; Case6:1.2D+1.6WX+L+0.5S and Case7: 1.2D+1.6WZ+L+0.5S where D is dead load, L is live load, S is snow load, EX and EZ are earthquake loads which are applied on X and Z global directions respectively, WX and WZ are the wind loads which are applied on X and Z global direction respectively. In addition, the top story drift constraints in *x* and *y* directions are restricted as the 3.9 cm. Inter-story drift limit is 1.14 cm for first story, and 0.915 cm for other stories. Maximum deflection of beam members is restricted as 2.03 cm.



a) 3D view

b) Plan view

c) Side view

Figure 2: 3D, plan and side views of four-story, 132-members space steel frame.

Table 2: Gravity loading on the beams of 132-members space steel frame.

	Beam Type	**Uniformly distributed load,(KN/m)** Outer Span	Inner Span
Load Case 1	Roof Beams	-7.012182404	-14.02436481
	Floor Beams	-8.180879472	-16.36175894
Load Case 2	Roof Beams	-7.932531345	-15.86506269
	Floor Beams	-18.25598315	-36.5119663
Load Case 3	Roof Beams	-9.957299014	-19.91459803
	Floor Beams	-10.52587014	-21.05174027
Load Case 4	Roof Beams	-7.012182404	-14.02436481
	Floor Beams	-10.52587014	-21.05174027
Load Case 5	Roof Beams	-7.012182404	-14.02436481
	Floor Beams	-10.52587014	-21.05174027
Load Case 6	Roof Beams	-6.763834278	-13.52766856
	Floor Beams	-14.2763359	-28.55267179
Load Case 7	Roof Beams	-6.763834278	-13.52766856
	Floor Beams	-14.2763359	-28.55267179

Table 3: Earthquake loading on the beams of 132-members space steel frame.

Floor Number	Earthquake Design Load (kN)	Floor Number	Earthquake Design Load (kN)
1	29.23	3	82.35
2	55.28	4	110.15

This example is run 20 times by AFFA, ABC and CS algorithms using different seed values in each design. Design parameters used in AFFA, ABC and CS algorithms, average weights, and standard deviations obtained by mentioned metaheuristics are illustrated in Table 4. The weights of the best designs for each algorithm are compared to the best designs obtained by each algorithm as well those obtained by using Dynamic Harmony Search and Ant Colony algorithms as given in [23]. W sections of the optimum designs and corresponding maximum constraint values for each algorithm are illustrated in Table 5. It is apparent from the Table 5 that the best design having the lightest weight is obtained by the Artificial Bee Colony (ABC) algorithm (594.00kN). The second best design is obtained by the Adaptive Firefly algorithm (AFFA). Difference between the optimum results is only 2.6%. The third and fourth best designs are attained by Dynamic Harmony Search (DHS) and Cuckoo Search (CS) algorithms. Although minimum weights reached by these designs are so close to each other (0.6%), average weight of these designs is 9.55% heavier than the best design obtained by ABC algorithm. Search histories of all solutions are illustrated in Figure 5.

Table 4: Design parameters used in AFFA, ABC and CS algorithms, average weights, and standard deviations.

AFFA	**ABC**	**CS**
Number of fireflies = 50	Number of employed bees = 25	Number of Nests =50
Maximum randomness parameter (α_{max})=0.8	Number of onlooker bees =25	Value of PA=0.35
Minimum randomness parameter (α_{min})=0.1	Controlling parameter adjusting food source (limit) = 200	
attractiveness at original location (β_0)=0.5	Maximum number of cycles = 1000	
Absorption coefficient (γ)=10		
Maximum iteration number = 50,000	Maximum iteration number = 50,000.	Maximum iteration number = 50,000
Average weight = 621.59 kN	Average weight = 611.55 kN	Aveage weight = 666.53kN
Std. Deviation =12.07	Std. Deviation =15.12	Std. Deviation =11.09

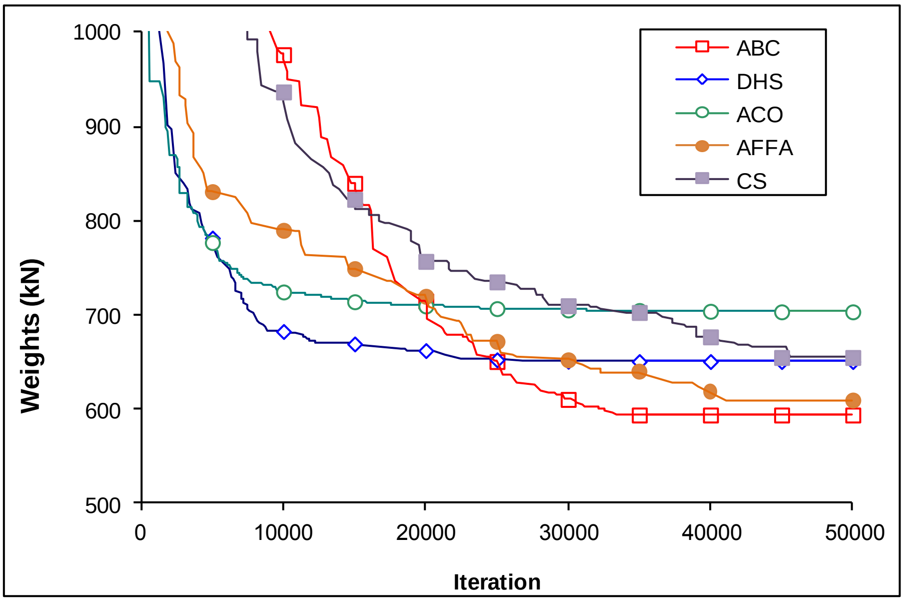


Figure 3: Search histories of four-story, 132-members space steel frame.

Table 5 Design results for four-story, 132-members space steel frame.

#	Group Type	AFFA	DHS	ACO	ABC	CS
1	Column	W250X49.1	W410X60	W250X58	W410X85	W200X86
2	Column	W250X49.1	W250X49.1	W920X223	W410X85	W360X101
3	Column	W310X79	W310X86	W760X185	W610X174	W250X115
4	Column	W310X117	W250X73	W760X220	W610X174	W310X129
5	Column	W310X97	W310X97	W530X300	W310X79	W360X179
6	Column	W360X147	W200X46.1	W920X365	W310X107	W360X179
7	Column	W200X46.1	W310X74	W310X60	W200X52	W410X132
8	Column	W200X46.1	W250X73	W690X384	W250X67	W760X134
9	Column	W460X74	W410X60	W530X92	W250X67	W360X134
10	Column	W460X74	W410X53	W610X92	W250X67	W530X138
11	Column	W360X179	W760X147	W760X314	W360X147	W360X216
12	Column	W360X179	W460X144	W1000X321	W360X147	W360X216
13	Column	W460X113	W410X75	W690X140	W310X97	W310X97
14	Column	W460X113	W360X72	W920X313	W310X97	W410X100
15	Column	W530X92	W310X60	W410X75	W610X113	W360X91
16	Column	W530X92	W250X58	W460X82	W760X134	W360X91
17	Column	W200X46.1	W410X67	W610X262	W360X101	W360X91
18	Column	W250X58	W410X60	W920X313	W760X134	W250X101
19	Column	W410X100	W410X75	W410X132	W310X107	W360X101
20	Column	W530X101	W310X67	W760X147	W310X107	W310X129
21	Column	W250X80	W310X60	W690X125	W310X107	W360X179
22	Column	W250X80	W250X58	W690X265	W530X123	W360X179
23	Column	W310X67	W200X59	W310X107	W460X74	W360X134
24	Column	W310X67	W200X46.1	W360X179	W690X125	W760X134
25	Column	W310X79	W310X86	W460X158	W460X158	W360X216
26	Column	W310X117	W310X79	W920X253	W460X158	W610X217
27	Column	W410X60	W250X49.1	W200X59	W200X59	W530X123
28	Column	W410X60	W200X46.1	W310X60	W410X60	W360X134
29	Beam	W530X74	W610X82	W460X60	W610X82	W530X74
30	Beam	W410X46.1	W410X53	W460X52	W460X52	W410X60
Min. weight(KN)		609.86	650.77	703.55	594.00	655.07
Max. top st. drift (cm)		2.95	2.83	2.99	3.05	2.87
Max. Inter-st. drift (cm)		1.05	1.05	0.88	1.14	0.87
Max. strength Constr.		0.974	0.999	0.895	0.936	0.889

7. Conclusions

The computational performance of three popular recent metaheuristic algorithms, namely FFA, ABC, and CS in the minimum weight design of space steel frame is investigated. 132- members space steel frame is designed by using these three different algorithms. The optimum designs obtained by FFA, ABC, and CS algorithms are compared by each other as well as by those attained by the Dynamic Harmony Search and Ant Colony Optimization algorithms. It is noticed that the lightest optimum design is attained by the artificial bee colony algorithm. Adaptive Firefly algorithm (AFFA) is the second bests. It is also noticed the performances of CS and DHS algorithms are close to each other in this particular problem. The optimum design obtained by Ant Colony Optimization algorithm is the heaviest weight among the other algorithms which is 8.73% heavier than the one determined by ABC algorithm.

8. References

[1] S. Gholizadeh, Optimum design of structures by an improved particle swarm algorithm, *Asian Journal of Civil Engineering*, 11 (6), 779–796, 2010.

[2] S. Kazemzadeh Azad and S. Kazemzadeh Azad, Optimum design of structures using an improved firefly algorithm, *International Journal of Optimization in Civil Engineering*, 1 (2), 327–340, 2011.

[3] S. Gholizadeh and H. Barati, A Comperative Study of Three Metaheuristics for Optimum Design of Trusses, *International Journal of Optimization in Civil Engineering*, 3 (3), 423–441, 2012.

[4] X-S. Yang, *Nature-inspired metaheuristic algorithms*, Luniver Press, UK, 2008.

[5] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Global Optimization*, 39, 459-471, 2007.

[6] B. Basturk and D. Karaboga, An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization, *IEEE Swarm Intelligence Symposium*, Indianapolis, Indiana, USA, 2006.

[7] D. Karaboga and B. Basturk, On the Performance of Artificial Bee Colony (ABC) Algorithm, *Applied Soft Computing*, 8 (1), 687–697, 2008.

[8] D. Karaboga and B. Basturk, *Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems*, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, 789–798, Springer-Verlag, 2007.

[9] X-S Yang and S. Deb, Engineering Optimization by Cuckoo Search, *International Journal of Mathematical Modeling and Numerical Optimization*, 1 (4), 330–343, 2010.
[10] LRFD-AISC, *Structural Members Specifications Codes, Volume 1, 3rd edition*, American Institute of Steel Construction 2001.
[11] I. Aydogdu and M.P. Saka, Ant colony optimization of irregular steel frames including elemental warping effect, *Advances in Engineering Software*, 44 (1), 150-169, 2012.
[12] Ad Hoc Committee on Serviceability, Structural Serviceability: A critical Appraisal and Research needs, *Journal of Structural Engineering*, ASCE, 112 (12), 2646-2664, 1986.
[13] M.P. Saka, *Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review*, in B.H.V. Topping (Ed.), Civil Engineering Computations: Tools and Techniques, Saxe-Coburg Publications, Stirlingshire, UK, 105-147, 2007.
[14] M.P. Saka and E. Dogan, *Recent developments in metaheuristic algorithms: a review*, in B.H.V. Topping (Ed.), Computational Technology Reviews, 5, 31-78, 2012.
[15] M.P. Saka, E. Dogan and I. Aydogdu, *Review and Analysis of Swarm-Intelligence Based Algorithms, in a book titled "Swarm Intelligence and Bio-Inspired Computation*, in Yang et.al. (Eds.), Elsevier, 2013.
[16] S. Carbas and M.P. Saka, Optimum Topology Design of Various Geometrically Nonlinear Latticed Domes Using Improved Harmony Search Method, *Structural and Multidisciplinary Optimization*, 45 (3), 377-399, 2012.
[17] L.S. Coelho and D.L.A. Bernert, An improved harmony search algorithm for synchronization of discrete-time chaotic systems, *Chaos, Solitons and Fractals*, 41 (5):2526–2532, 2009.
[18] R.N. Mantegna, Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes, *Physical Review E*, 49 (5), 4677-4683, 1994.
[19] S. Carbas, I. Aydogdu and M.P. Saka, Optimum Design of Steel Frames Using Adaptive Harmony Search Method to LRFD-AISC, *8th World Congress on Structural and Multidisciplinary Optimization*, Lisbon, Portugal, 2009.
[20] I. Aydogdu, A. Akin and M.P. Saka, Optimum Design of Steel Space Frames by Artificial Bee Colony Algorithm, *10th Congress on Advances in Civil Engineering*, Ankara, Turkey, 2012
[21] I. Aydogdu, A. Akin and M.P. Saka, Discrete Design Optimization of Steel Space Frames using the Adaptive Firefly Algorithm, *Proceedings of The Eleventh International Conference on Computational Structures Technology*, Dubrovnik, Croatia, 2012.
[22] ASCE 7-05, *Minimum Design Loads for Building and Other Structures*, 2005.
[23] M.P. Saka, I. Aydogdu, O. Hasancebi and Z.W. Geem, Harmony Search Algorithms in Structural Engineering, *Computational Optimization and Applications in Engineering and Industry, Studies in Computational Intelligence*, 359, 145-182, 2011.