Steel and Composite Structures, Vol. 12, No. 6 (2012) 505-522
DOI: http://dx.doi.org/10.12989/scs.2012.12.6.505

Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm

S. O. Degertekin

Department of Civil Engineering, Dicle University, 21280, Diyarbakir, Turkey

(Received September 06, 2011, Revised March 27, 2012, Accepted March 30, 2012)

Abstract. An artificial bee colony (ABC) algorithm is developed for the optimum design of geometrically non-linear steel frames. The ABC is a new swarm intelligence method which simulates the intelligent foraging behaviour of honeybee swarm for solving the optimization problems. Minimum weight design of steel frames is aimed under the strength, displacement and size constraints. The geometric non-linearity of the frame members is taken into account in the optimum design algorithm. The performance of the ABC algorithm is tested on three steel frames taken from literature. The results obtained from the design examples demonstrate that the ABC algorithm could find better designs than other meta-heuristic optimization algorithms in shorter time.

Keywords: artificial bee colony algorithm; geometrically non-linear steel frames; optimum design; load and resistance factor design.

1. Introduction

Many meta-heuristic search methods have been developed for the optimum design of steel frames in recent years. Genetic algorithms (GAs), ant colony optimization (ACO), particle swarm optimization (PSO) and harmony search (HS) are among the most popular meta-heuristic search methods. The main character of these methods is that they employ a population of individuals to optimize the problem.

GAs are based on evolution theory of Darwin's (Holland 1975) which applies the principle of survival of robust individuals and the elimination of the others in a population. GAs has been used for optimum design steel frames by many researchers (Camp *et al.* 1998, Pezeshk *et al.* 2000, Toropov and Mahfouz 2001, Hayalioglu and Degertekin 2005, Liu *et al.* 2006, Degertekin *et al.* 2008a).

ACO and PSO are the most popular optimization methods in the swarm intelligence. ACO simulates the positive feedback process exhibited by a colony of ants (Dorigo 1992) and PSO bases on the social behaviour of swarms (Kennedy and Eberhart 1995). Both methods have been applied to the optimum design of steel frames (Camp *et al.* 2005, Kaveh and Shojaee 2007, Hasancebi *et al.* 2009, 2010).

HS bases on the analogy between the performance process of natural music and searching for solutions to optimization problems. The method is proposed by Geem *et al.* (2001). It also applied to the optimal design of steel frames (Degertekin 2008a, 2008b, Saka 2009, Hasancebi *et al.* 2009, 2010).

The ABC is a new swarm intelligence method which simulates the intelligent foraging behaviour of

* Corresponding author, Ph.D., E-mail: sozgurd@gmail.com

honeybee swarm for solving the optimization problems. The method is originally introduced by Karaboga (2005). Although it is relatively new meta-heuristic search method, it has been applied in a variety of optimization problems.

Karaboga and Basturk (2007) developed an ABC algorithm for optimizing multivariable functions. The ABC algorithm was tested on five high dimensional numerical benchmark functions and the results obtained by the ABC algorithm were compared with those of other meta-heuristic methods. The results obtained by the study revealed that the proposed the ABC algorithm could be efficiently used for multivariable, multimodal function optimization.

Karaboga (2009) introduced an ABC algorithm for designing low and high order digital recursive filters. The results obtained by the ABC were compared with the results obtained by a conventional optimization algorithm and particle swarm optimization. It was observed from the results that the ABC algorithm seemed as an alternative optimization algorithm for designing low and high order digital recursive filters.

Singh (2009) proposed an ABC algorithm for the leaf-constrained minimum spanning tree problem. The ABC algorithm is compared with three other meta-heuristic approaches; a genetic algorithm, an ant colony optimization and a tabu search approach. The proposed ABC algorithm outperformed all the other algorithms.

An ABC algorithm was applied to solve the economic load dispatch problem by Hemamalini and Simon (2010). The performance of ABC algorithm was demonstrated by using some test cases. They compared the results with other population-based methods such as genetic algorithm, artificial immune system, particle swarm optimization and evolutionary programming. From the results obtained, it was concluded that ABC algorithm is a promising technique for solving complex non-smooth optimization problems.

Omkar *et al.* (2011) developed a model for multi-objective design optimization of laminated composite components based on vector evaluated ABC algorithm. The method was validated for a number of different loading configurations, unaxial, biaxial and bending loads. The performance of the ABC algorithm was evaluated in comparison with other techniques like particle swarm optimization, artificial immune system and genetic algorithm. The results of ABC were at par with that of those algorithms for all loading configurations.

Sonmez (2011a) presented an ABC algorithm for the discrete optimum design of truss structures. The results obtained by this study showed that the ABC algorithm is very effective method for discrete optimization designs of truss structures.

Sonmez (2011b) proposed an ABC algorithm with an adaptive penalty function approach (ABC-AP) in order to minimize the truss structures. The results obtained from this study demonstrated that ABC-AP algorithm provides results as good as or better than other heuristic search algorithms. However, ABC-AP shows very poor convergence capability compared with other heuristic algorithms.

A modified ABC algorithm for constrained optimization problems is proposed by Karaboga and Akay (2011). For constraint handling, ABC algorithm uses Deb's rules (Deb 2000) consisting of three simple heuristic rules and a probabilistic selection scheme for feasible solutions based on their fitness values and infeasible solutions based on their violation values. Moreover, a statistical parameter analysis of the modified ABC algorithm is conducted and appropriate values for each control parameter are obtained using analysis of the variance and analysis of mean statistics. The performance of the modified ABC algorithm is compared with those of state-of-the-art algorithms for a set of constrained test problems. The numerical results shows that ABC algorithm can be efficiently used for solving constrained optimization problems.

The literature survey reveals that although the ABC algorithm has been applied for different optimization problems in the last several years, it only has been applied for the sizing optimization of truss structures in the field of structural optimization.

This study aims at introducing the ABC algorithm into the optimum design of a different structural system, i.e., geometrically non-linear steel frames, under the actual design constraints of American Institute of Steel Construction (AISC)-Load and Resistance Factor Design (LRFD) specification (AISC-LRFD 2001). Discrete design variable selected from the standard set of AISC wide-flange (W) shapes are used in the optimal design of frames. The efficiency of the ABC algorithm is verified by using three steel frames taken from literature. The results obtained using the ABC algorithms are compared with the other meta-heuristic search algorithms.

The remainder of this paper is organized as follows: the formulations of optimum design problem are described in Section 2. The artificial bee colony algorithm is defined in Section 3; Optimum design of geometrically non-linear steel frames using the ABC algorithm is explained in Section 4. The design examples taken from the literature are presented in Section 5. Finally, the results obtained from the design examples are discussed in Section 6.

2. The formulations of optimum design problem

The minimum weight design problem for a frame structure can be formulated as

$$\text{Minimize} \quad W(I) = \sum_{k=1}^{ng} A_k \sum_{i=1}^{mk} \rho_i L_i \tag{1}$$

where I is the vector containing the design variables $I = I_1, I_2, ..., I_{ng}$, $I_i \in I_{sl}$, $(i = 1, 2, \ldots, ng)$, I_{sl} is the standard set of AISC wide-flange (W) shapes, ng is the total numbers of groups (i.e. number of design variables) in the frame, mk is the total numbers of members in group k, ρ_i and L_i are the density and the length of member i, A_k is the cross-sectional area of member group k. A design variable in the optimization problem denotes a member group in a steel design. Design variables are selected from a section list and each of section is represented by a sequence number in that list (Degertekin *et al.* 2008b). The unconstrained objective function could be expressed as

$$\varphi(I) = W(I)[1 + C \times \kappa] \tag{2}$$

where C is the constraint violation function, κ is the penalty function factor. The constraint violation function is expressed as

$$C = \sum_{j=1}^{m}\sum_{l=1}^{nl} C_{jl}^{d} + \sum_{j=1}^{ns}\sum_{i=1}^{nsc}\sum_{l=1}^{nl} C_{jil}^{is} + \sum_{k=1}^{nj} C_{k}^{sb} + \sum_{n=1}^{ncl} C_{n}^{sc} + \sum_{i=1}^{nm}\sum_{l=1}^{nl} C_{il}^{i} \tag{3}$$

where C_{jl}^{d}, C_{jil}^{is}, C_{k}^{sb}, C_{n}^{sc} and C_{il}^{i} are the constraints violations for displacement, interstorey drift, size constraint for beams, size constraint for columns and interaction formulas of AISC-LRFD (2001) specification, respectively. m and nl are the number of restricted displacements and the total number of loading conditions. ns and nsc are the number of storeys in the frame and the number of columns in a storey, nj and ncl are the total number of joints in the frame excluding supports and the total number of columns in the frame except the ones at the bottom floor, nm is the is total number of members in the frame. The constraint violation C is defined as follows

$$C_i = \begin{cases} 0 & if \quad \beta \le 0 \\ \beta_i & if \quad \beta_i > 0 \end{cases} \tag{4}$$

The displacement and interstorey drift constraints are imposed on the steel frames

$$\beta_{ij}(I) = \frac{|\delta_{jl}|}{|\delta_{ju}|} - 1 \,, j = 1, \text{.......}, m, \; l = 1, \text{.......}, nl \tag{5}$$

$$\beta_{jl}(Z) = \frac{|\Delta_{jil}|}{|\Delta_{ju}|} - 1 \,, j = 1, \text{.......}, ns, \; i = 1, \text{.......}, nsc, \; l = 1, \text{.......}, nl \tag{6}$$

where δ_{jl} is the displacement of the *j-th* degree of freedom due to loading condition l, δ_{ju} is its upper bound, Δ_{jil} is interstorey drift of *i*-th column in the *j*-th storey due to loading condition l, Δ_{ju} is its limit.

The size constraints are given as

$$\beta_k(I) = \frac{b_{fbk}}{b_{fck}} - 1.0 \,, k = 1, \text{.......}, nj \tag{7}$$

$$\beta_n(I) = \frac{d_{un}}{d_{bn}} - 1.0 \,, n = 1, \text{.......}, ncl \tag{8}$$

where b_{fbk} and b_{fck} are the flange widths of the selected beam and column respectively. d_{un} and d_{bn} are depths of steel sections selected for upper and lower floor columns.

The strength constraints are expressed in the following interaction equations (AISC-LRFD 2001).

For members subject to bending moment and axial force

for $\dfrac{P_u}{\phi P_n} \ge 0.2$

$$\beta_{il}(I) = \left(\frac{P_u}{\phi P_n}\right)_{il} + \frac{8}{9}\left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right)_{il} - 1.0 \quad i = 1, ..., nm, \; l = 1, \text{.....}, nl \tag{9}$$

for $\dfrac{P_u}{\phi P_n} < 0.2$

$$\beta_{il}(I) = \left(\frac{P_u}{2\phi P_n}\right)_{il} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}}\right)_{il} - 1.0 \quad i = 1, ..., nm, \; l = 1, \text{.....}, nl \tag{10}$$

where P_u is the required axial strength (compression or tension), P_n is the nominal axial strength (compression or tension), M_{ux} is the required flexural strengths about the major axis including second-order effects (geometric non-linearity), M_{uy} is the required flexural strengths about the minor axis (for two-dimensional frames, M_{uy} is 0), M_{nx} is the nominal flexural strength about the major axis, M_{ny} is the nominal flexural strength about the minor axis, $\phi = \phi_c$ is the resistance factor for compression (equal to 0.85), $\phi = \phi_t$ is the resistance factor for tension (equal to 0.90), ϕ_b is the flexural resistance factor (equal to 0.90).

If axial force is compression in Eqs. (9) and (10), P_u is the required compressive strength, P_n is the nominal compressive strength, $\phi = \phi_c$ is the resistance factor for compression (equal to 0.85). The explanation for the other terms is the same as the previous paragraph. The nominal compressive strength of a member is computed as

$$P_n = A_g \cdot F_{cr} \tag{11}$$

$$F_{cr} = (0.658^{\lambda_c^2})F_y \qquad \text{for } \lambda_c \le 1.5 \tag{12}$$

$$F_{cr} = \left(\frac{0.877}{\lambda_c^2}\right)F_y \qquad \text{for } \lambda_c > 1.5 \tag{13}$$

$$\lambda_c = \frac{KL}{r\pi}\sqrt{\frac{F_y}{E}} \tag{14}$$

where A_g is cross-sectional area of member; K is effective length factor; E is modulus of elasticity; r is governing radius of gyration; L is member length; and F_y is yield stress of steel. The effective length factor (K) for unbraced frames is calculated from the following approximate equations (Dumonteil 1992)

$$K = \sqrt{\frac{1.6G_AG_B + 4.0(G_A + G_B) + 7.50}{G_A + G_B + 7.50}} \tag{15}$$

where subscripts A and B denote the two ends of the column under consideration. The restraint factor G is stated as

$$G = \frac{\sum(I_c/L_c)}{\sum(I_g/L_g)} \tag{16}$$

where I_c is the moment of inertia and L_c is the unsupported length of a column section; I_g is the moment of inertia of a girder and L_g is unsupported length of a girder. Σ indicates a summation for all members rigidly connected to that joint (A or B) and lying in the plane of buckling of the column under consideration.

The non-linear analysis of steel frames takes into account the geometric non-linearity of frame members. For this purpose, a plane-frame member is modified to include geometric non-linearity effect (P-Δ effect). The structure stiffness matrix is constructed by superimposing the member stiffness matrices contain geometric non-linearity. The steps of this iterative process is explained in detail in author's previous work (Hayalioglu and Degertekin 2005) and not repeated here.

3. The artificial bee colony algorithm

The ABC algorithm is a colony-based meta-heuristic search method originally proposed by Karaboga (2005). The method imitates the intelligent behaviour of honeybee swarm. Every bee colony consists of three types of bee which are employed bees, onlooker bees and scout bees. Employed bees make up half of the colony and onlooker bees make up the other half. If an employed or onlooker bee can not find a better food source for a while, it turns into a scout bee. The employed bees visit a food source within the neighbourhood of the food source in their memory and carry the loads of nectar from the food source to the hive. Then, they share their information about the food source by dancing in the hive. The more they have nectar, the more they dance. Onlooker bees observe the dance of the employed bees in the hive and choose a food source according to the nectar amount of the food source. The food sources, which have more nectar, are visited by more bees than the ones which have less nectar. The scout bees are explorer bees which look for a new food source in the vicinity of the hive randomly. If a food source is exploited entirely, the employed bee which exploits that food source becomes a scout bee (Karaboga and Basturk 2008). The employed and onlooker bees exploit the available food sources while scout bees are exploring the new ones.

The ABC algorithm consists of three stages. At the first stage, an initial bee colony is generated randomly. In this process, a set of food sources is randomly selected by the bees and their nectar amounts are determined. Each food source I_i in the colony ($i = 1, 2, \ldots, ntb$) is represented by a *ng*-dimensional vector. Here, *ntb* is the number of food sources (i.e., the number of artificial bee) and *ng* denotes the number of design variables. The food sources in the colony are sorted according to their nectar amounts.

At the second stage, an employed bee which is in the position *i* changes its position in order to find a new food source as shown in the following equation

$$I_{ij}^{new} = I_{ij}^{old} + INT[\phi_{ij}(I_{ij}^{old} - I_{kj})] \tag{17}$$

where i, $k \in \{1, 2, ..., neb\}$ and $j \in \{1, 2, ..., ng\}$ are the randomly selected indexes, *i* has to be different from *k, neb* is the number of employed bees, *j* is the design variable, ϕ_{ij} is a random number which is uniformly distributed over the interval [-1,1]. After that, the employed bee evaluates the nectar amount of the new food source. If the nectar amount is higher than that of the previous one, the bee moves on new one. After all employed bees complete their search process, they share the nectar information of the food sources and their position with the onlooker bees on the dance area. The onlooker bees evaluate the nectar information given by the employed bees and choose a food source depending on the probability in the following way (Karaboga and Basturk 2007)

$$p_i = \frac{\varphi(I)_i}{\sum_{n=1}^{neb} \varphi(I)_n} \tag{18}$$

in which $\varphi(I)_i$ is the objective function value of the solution *i.* This value is proportional to the nectar amount of the food source in the position *i*. After the probability values of the employed bees have been determined, the onlooker bee changes the selected employed bee's position as follows

$$I_{lj}^{new} = I_{lj}^{old} + INT[\phi_{lj}(I_{lj}^{old} - I_{lj})] \tag{19}$$

where $l \in \{1, 2, ..., nob\}$ and $j \in \{1, 2, ..., ng\}$ are the randomly selected indexes, ϕ_{lj} is a random number which is uniformly distributed over the interval [-1,1] and *nob* is the number of onlooker bees. If the nectar amount of the new source is higher than that of the previous one, the onlooker bee selects the new position; otherwise, it remains in its previous position. This provides a greedy selection strategy between the old food source and the new one and it is repeated for each onlooker bee.

At the third stage: if a bee in the colony could not improve its nectar amount in a predetermined number of trials, the food source exploited by this bee is deserted and that bee becomes a scout bee. The predetermined number of trials is equal to the limit value (*lv*). The aim of using the limit value is to provide a new search space in the ABC algorithm. If the food source I_{ij} is deserted, a new food source is explored by scout bee as stated in the following equation (Hemamalini and Simon 2010)

$$I_{ij} = I_{j\,min} + INT[rand(0,1)(I_{j\,max} - I_{j\,min})] \qquad j = 1, 2, ..., ng \tag{20}$$

where $I_{j\,min}$ and $I_{j\,max}$ are the minimum and maximum limit values of *j-th* design variable. An iteration is finished when the three stages are completed. The ABC algorithm repeats the search process until a terminating criterion is satisfied.

In the classical approach, the steps are repeated until the predetermined total number of iterations is performed or current optimum design remains the same during the execution of the predetermined value of iterations of the maximum iteration number. It is obvious that this approach bases on the computational experience which is obtained after numerous executions. For this reason, the following termination criterion is implemented (Lamberti and Pappalettere 2009)

$$\frac{STD[\varphi(I^1),\, \varphi(I^2),\, \ldots\ldots,\, \varphi(I^{NTB})]}{\sum_{n=1}^{ntb} 1/\varphi(I^n)} \le \varepsilon_{conv} \tag{21}$$

where $\varepsilon = 10^{-7}$ and *ntb* is the total number of bees in the colony. Eq. (21) states that the ratio between the standard deviation of objective functions and the average value of objective function for the designs stored in the bee colony must be less than 10^{-7}.

There are two control parameters in the ABC algorithm: the total number of bees in the colony (*ntb*) which is the sum of the number of employed bees (*neb*) and the number of onlooker bees (*nob*), the predetermined limit value (*lv*).

Selection processes performed by the ABC consist of four steps (Karaboga and Akay 2009): (i) A global probabilistic selection process, in which the probability value given in Eq. (18) is used by the onlooker bees to discover promising search areas. (ii) A local probabilistic selection process performed by the employed bees and onlookers to determine a food source around the current food source using Eqs. (17) and (19). (iii) A greedy selection process performed by the employed and onlooker bees. They forget the previous food source and memorize the new one, if the nectar amount of the new one is better than the previous one. (iv) A random exploration stage is performed by scout bees as shown in Eq. (20).

An analogy between the ABC algorithm and the optimum design of steel frames is established in the following way: an employed bee or onlooker bee in the colony denote a possible frame design in the optimization process. The nectar amount of a food source represents the objective function value of the design.

4. Optimum design of geometrically non-linear steel frames using the ABC algorithm

The proposed ABC algorithm for optimum design of geometrically non-linear analysis of steel frames could be explained in the following steps:

Step 1 : Initialize the ABC algorithm parameters: the total number of bees in the colony (*ntb*), the predetermined limit value (*lv*). Set the iteration counter: $it = 0$.

Step 2 : Randomly generate an initial colony which consist of frame designs. Each design variable in a frame is generated between the first and the last AISC wide-flange (W) shapes in the standard section list using Eq. (22)

$$I_i^j = I_{i\,min}^j + INT[rand(0,1)(I_{i\,max}^j - I_{i\,max}^j)] \qquad i = 1, 2, \ldots, ng, \qquad j = 1, 2, \ldots, ntb \tag{22}$$

The frame designs in the colony could be shown explicitly in the following matrix form

$$I_i^j = \begin{bmatrix} I_1^1 & I_2^1 & . & . & I_{ng}^1 \\ I_1^2 & I_2^2 & . & . & I_{ng}^2 \\ . & . & . & . & . \\ . & . & . & . & . \\ I_1^{ntb} & I_2^{ntb} & . & . & I_{ng}^{ntb} \end{bmatrix} \begin{array}{l} \rightarrow \varphi(I^1) \\ \rightarrow \varphi(I^2) \\ \rightarrow . \\ \rightarrow . \\ \rightarrow \varphi(I^{ntb}) \end{array} \qquad i = 1, 2, ..., ng, \quad j = 1, 2, ..., ntb \tag{23}$$

Each row in the matrix denotes a frame design in the matrix. I^1, I^2, ..., I^{ntb} and $\varphi(I^1)$, $\varphi(I^2)$, ..., $\varphi(I^{ntb})$ are the frame designs and the corresponding unconstrained objective function value, respectively. The steel designs in the matrix are sorted by the objective function values in ascending form (i.e. $\varphi(I^1) < \varphi(I^2) < ...... < \varphi(I^{ntb})$. Penalized designs can be also included in the matrix. The first half of the frame designs is assigned as employed bees (*neb*) and the second half of the frame designs are appointed as onlooker bees (*nob*).

Step 3 : Analyze the frames, obtain the nodal displacements and member forces. Calculate the objective function $\varphi(I)$ using Eq.(2). Sort the frame designs according to their $\varphi(I)$ values in ascending form. In this case, the best design with the lowest objective function value is placed in the first row of the colony and the worst design (I^{worst}) with the highest objective function value ($\varphi(I^{worst})$) is placed in the last row of the colony.

Step 4 : Increase the iteration counter, $it = it + 1$.

Step 5 : Randomly select a frame design with the value of $\varphi(I^r)$ objective function (I^r; $r = 1, 2, ….,$ *neb*) from the first half of the colony. Produce a modification on the selected design for finding a new one. The new design is obtained by altering the value of all design variables using Eq. (17). Analyze the new frame, obtain its response and the value of objective function ($\varphi(I^{new})$).

Step 6 : If $\varphi(I^{new})<\varphi(I^{worst})$, the new design is included in the colony and the worst one is excluded from the colony. Sort the frames according to their $\varphi(I)$ values.

Step 7 : Repeat the steps 5 and 6 until all designs in the first half of the colony are selected once.

Step 8 : Share the designs obtained from steps 5 and 6 with the ones in the second half of the colony. A design from the first half of the colony is chosen with a probability p_i. The Eq. (18) is suitable for maximization problems. Nonetheless, the aim of the optimization problem presented in this study is to minimize the weight of the frame structures. For this purpose, the Eq. (18) is modified as follows

$$p_i = \frac{1/\varphi(I^i)}{\sum_{n=1}^{neb} 1/\varphi(I^n)} \qquad i = 1, 2, \cdots, neb \tag{24}$$

Step 9 : Produce the modification on the position of the frame design using Eq.(19) and obtain a new frame. If the value of $\varphi(I)$ for the new frame design is less than the worst one in the colony, it is replaced with the worst one. Otherwise, it is retained. Sort the designs according to their $\varphi(I)$ values in ascending form again.

Step 10 : Repeat the steps 8 and 9 until all designs in the second half of the colony are selected once.

Step 11 : If a frame design except the best frame design in the colony is not improved in a predetermined limit value (lv), abandon this design. Generate a frame design using Eq. (20). Record the feasible design with the lowest objective function value and assign it as the current optimum.

Step 12 : Repeat the steps 4-11 until the Eq. (21) is satisfied. If it is satisfied, terminate the algorithm and define the current optimum as the final optimum design.

5. Design examples

The numerical efficiency of the ABC algorithm is investigated by solving weight minimization problems of three steel frames. The results obtained by the ABC algorithm are compared with those of the GA (Pezeshk *et al.* 2000, Hayalioglu and Degertekin 2005) and harmony search (HS) algorithm (Degertekin and Hayalioglu 2010).

The ABC algorithm parameters are not assigned directly by the user. Sensitivity analysis is carried out for the ABC algorithm in order to find the best combination of the ABC internal parameters.

Since the proposed algorithm is stochastic in nature, 20 independent optimization runs are performed for frame designs. The best and the worst designs obtained by the ABC algorithm along with the number of structural analyses required to complete the optimization process are reported in the following tables. The top storey drift and maximum inter-storey drift computed for the 20 independent runs also are specified. The maximum interaction ratio eventually present at the optimum design also is reported in the tables. The Young's modulus and the yield stress of steel members are taken as E = 200 GPa and f_y = 248.2 MPa, respectively. The ABC algorithm is coded in FORTRAN language. Optimization runs are executed on a Intel Pentium Core 2 Duo 2.2 GHz computer.

Because of the stochastic nature of the ABC algorithm, random number sequences are generated during the search process. Random numbers always produce the same number for different executions if the same SEED value is used in each run. In order to provide exact comparison in the 20 runs, the same SEED value is used for the initial population in each run.

5.1 Design of two-bay, three-storey frame

The two-bay three-storey frame shown in Fig. 1 is the first design example. This frame was optimized using GA (Pezeshk *et al.* 2000) under a single load case shown in Fig. 1.

Displacement and size constraints are not imposed for the design. The beam members are selected from a list with 267 W sections and the maximum depth of the column sections are restricted to the 25.4 cm. The member effective length factors K_x is calculated from the approximate equation proposed by Dumonteil (1992). For each column, the out-of-plane effective length factor (K_y) is considered as 1.0. The out-of-plane effective length factor for each beam member is specified to be 0.167. The length

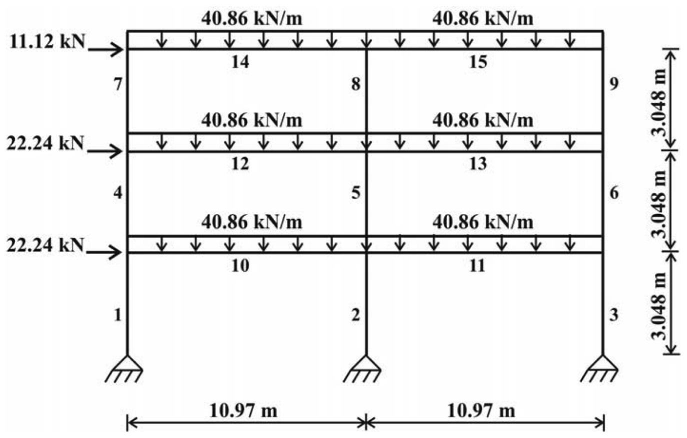

Fig. 1 Two-bay, three-storey frame

of the unbraced compression flange for each column member is directly calculated during the design process while the length of the unbraced compression flange for each beam member is specified to be 0.167 of the span length (Pezeshk *et al.* 2000). The penalty function factor remained constant at $k = 1$.

The results of the sensitivity analysis carried out to obtain the appropriate values of the *ntb* and *lv* in the ABC algorithm are presented in Table 1. It can be seen that the minimum weight design for the ABC algorithm is found by setting *ntb* = 60 and *lv* = 500.

Table 2 compares the results obtained by the ABC algorithm with the GA algorithm (Pezeshk *et al.* 2000). The ABC algorithm yielded 3.7% lighter design than that of the GA (Pezeshk *et al.* 2000). It is also noted that the ABC algorithm obtained its best design in 65% of the 20 runs. The average weight of 20 runs is calculated as 84.71 kN, with a standard deviation of 1.53 kN.

Fig. 2 provides the interaction ratio for the each member at the optimum design. It is seen from Fig. 2 that interaction ratio of four member is within 90% of maximum interaction ratio at the optimum. Fig. 3 shows the convergence curves of the ABC algorithm and the GA (Pezeshk *et al.* 2000). It is obvious that the ABC algorithm required slightly less structural analyses than the GA algorithm (Pezeshk *et al.* 2000) and the convergence capability of the ABC is more powerful than that of the GA's.

Table 1 Results of sensitivity analysis carried out to find the best combination of internal parameters of the ABC algorithm for the three-storey two-bay frame

Cases	*ntb*	*lv*	Weight (kN)
1	20	500	96.50
2	20	1000	96.50
3	20	1500	92.96
4	40	500	86.79
5	40	1000	86.79
6	40	1500	92.96
7	60	500	83.59
8	60	1000	86.79
9	60	1500	86.79
10	80	500	92.96
11	80	1000	92.96
12	80	1500	96.50

Table 2 Optimum results obtained for the two-bay three-storey frame

Member group	GA	ABC_{worst}	ABC_{best}
	Pezeshk *et al.* (2000)	This study	
Beam	W 24×62	W 24×62	W 24×62
Column	W 10×68	W 10×68	W 10×60
Weight (kN)	86.79	86.79	83.59
Top storey drift (cm)	*	1.37	1.52
Max. inter-storey drift (cm)	*	0.93	1.05
Max. interaction ratio	*	0.95	0.99
Number of structural analyses	1800	1380	1620

*Not available

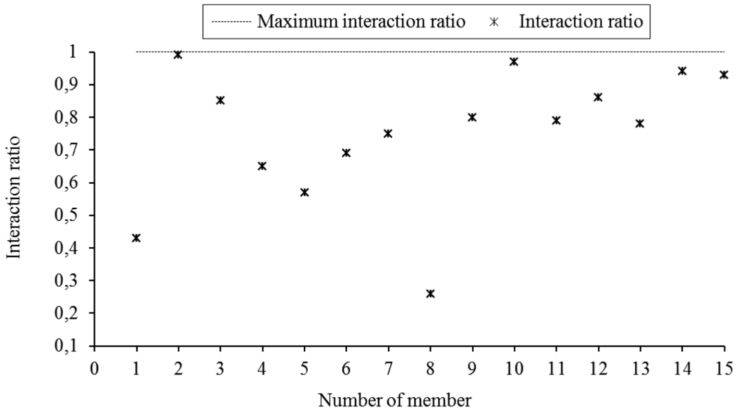

Fig. 2 Interaction ratios for the two-bay, three-storey frame

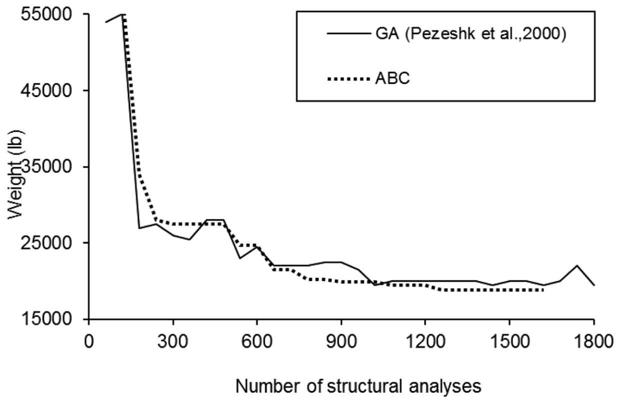

Fig. 3 Convergence curves for the two-bay, three-storey frame

5.2 Design of one-bay, ten-storey frame

Fig. 4 shows the frame configuration, dimensions, loading and numbering of grouping of members. This frame was optimized by Pezeshk *et al.* (2000) using GA under a single load case shown in Fig. 4.

The strength constraints taken from AISC-LRFD (2001) specification are used and a interstorey drift constraint is imposed as: interstorey drift < storey height/300. Beam element groups are chosen from 267 W-sections and five column groups are selected from only W14 and W12 sections. The effective length factors of members are calculated as $K_x \geq 1$ using the approximate equation proposed by Dumonteil (1992), whereas the out-of-plane length factor K_y is assigned as 1. For each beam member, the out-of-plane effective length factor is specified to be $K_y = 0.2$ i.e., floor stringers at 1/5 points of the span.

The results of the sensitivity analysis carried out to find the best combination of internal parameters for the ABC algorithm are presented in Table 3. The same values of internal parameters as in the three-storey two-bay frame are found also for the ten-storey one-bay frame.

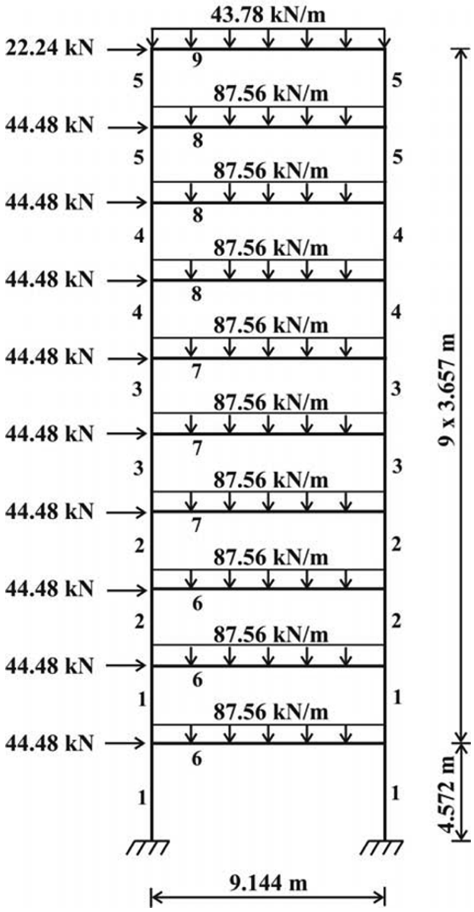

Fig. 4 One-bay, ten-storey frame

Table 3 Results of sensitivity analysis carried out to find the best combination of internal parameters of the ABC algorithm for the one-bay ten-storey frame

Cases	*ntb*	*lv*	Weight (kN)
1	20	500	315.97
2	20	1000	322.08
3	20	1500	322.08
4	40	500	313.06
5	40	1000	313.06
6	40	1500	319.92
7	60	500	308.68
8	60	1000	313.06
9	60	1500	313.06
10	80	500	319.92
11	80	1000	315.97
12	80	1500	319.92

Table 4 Design results for the one-bay, ten-storey frame

Group number	GA Pezeshk *et al.* (2000)	ABC_{worst}	ABC_{best}
		This study	
1	W 14×233	W 14×233	W 14×233
2	W 14×176	W 14×176	W 14×176
3	W 14×132	W 14×159	W 14×145
4	W 14×99	W 14×99	W 14×99
5	W 12×65	W 14×68	W 12×58
6	W 36×150	W 36×150	W 36×150
7	W 33×130	W 33×118	W 33×118
8	W 27×94	W 27×94	W 27×84
9	W 16×50	W 24×68	W 24×68
Weight (kN)	313.13	317.595	308.68
Top storey drift (cm)	*	9.54	9.98
Max. inter-storey drift (cm)	*	1.23	1.23
Average weight (kN)	*	314.153	
Standard deviation (kN)	*	3.72	
Max. interaction ratio	*	0.94	0.98
Number of structural analyses	3000	2820	2880

*Not available

Table 4 compares optimization results with the corresponding data reported in literature. It can be seen that the ABC found 1.4% lighter design than that of the GA (Pezeshk *et al.* 2000). It is interesting to note that the ABC algorithm obtained its best design in 55% of the 20 runs. For 20 runs, the average weight is obtained as 314.153 kN, with a standard deviation of 3.72 kN.

The highest interaction ratio for each group at the optimum design is shown in Fig. 5. It is apparent that the highest interaction ratio for the four member group is within 90% of maximum interaction ratio at the optimum.

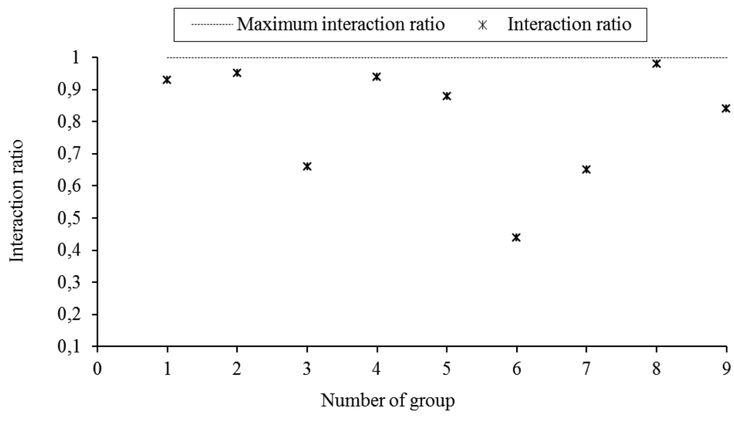

Fig. 5 The highest interaction ratio for each group for the one-bay ten-storey frame

Interstorey drifts also are demonstrated in Fig. 6. The worthwhile result obtained from the ABC algorithm that the interstorey drift is within 90% of the maximum interstorey drift between stories 2 to 6 and 7 to 8 as shown in Fig. 6. This indicates that interstorey drift constraint is dominant at the optimum.

The convergence curve for the best design is shown in Fig. 7. The ABC converged to the optimum after 2880 structural analyses which are slightly less than that of the GA (Pezeshk *et al.* 2000).

5.3 Design of four-bay, ten-storey frame

The last example is the four-bay, ten-storey frame shown in Fig. 8. This frame was optimized using the GA (Hayalioglu and Degertekin 2005) and the HS (Degertekin and Hayalioglu 2010).

The structure is divided into 12 groups of elements as shown in Fig. 5. Four different types of loads are employed: dead load (D), live load (L), roof live load (L_r), and wind loads (W). The following loads are applied: D = 30.65 kN/m, L = 21.37 kN/m, L_r = 12.26 kN/m, W = 34.22 kN. Four load combinations are considered, per AISC LRFD specification: I: ($1.4D$), II: ($1.2D + 1.6L + 0.5L_r$), III: ($1.2D + 1.6L_r + 0.5L$) , and IV: ($1.2D + 1.3W + 0.5L + 0.5L_r$). The allowable sway of the top storey must not exceed

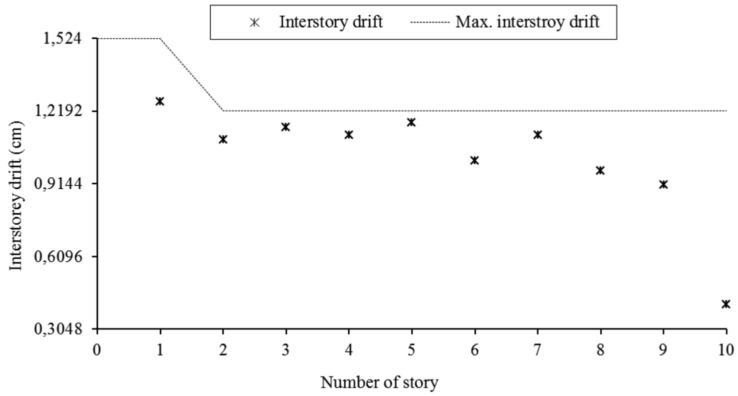

Fig. 6 Interstorey drifts for the one-bay ten-storey frame

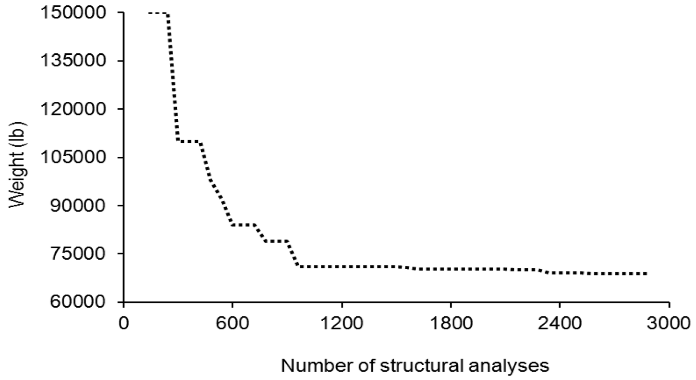

Fig. 7 Convergence curve for the one-bay, ten-storey frame

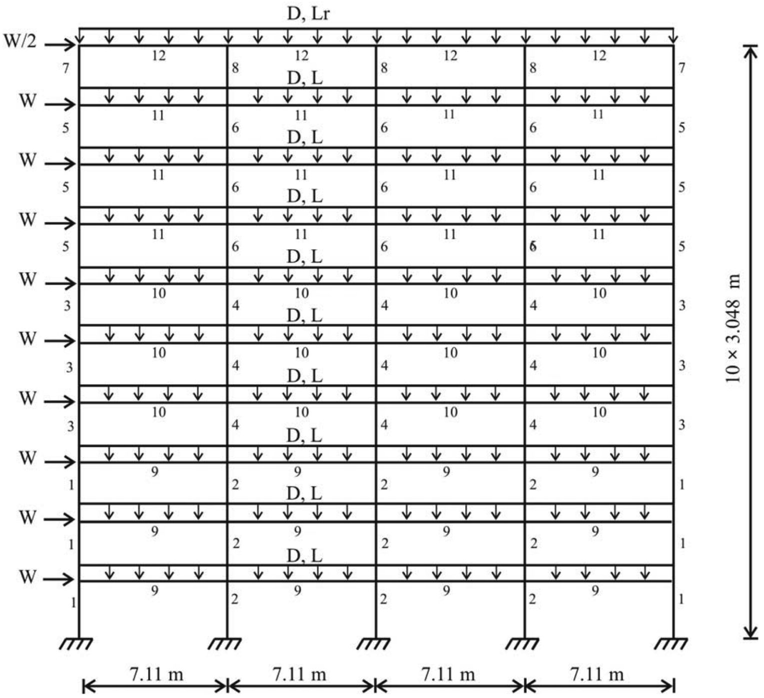

Fig. 8 Four-bay, ten-storey frame

Table 5 Designs results for four-bay, ten-storey frame

Group no.	GA Hayalioglu and Degertekin (2005)	HS Degertekin and Hayalioglu (2010)	ABC_{worst}	ABC_{best}
			This study	
1	W 24×104	W 24×104	W 24×101	W 21×101
2	W 27×146	W 27×146	W 27×147	W 21×147
3	W 16×77	W 18×86	W 18×76	W 14×68
4	W 21×122	W 21×101	W 18×106	W 18×97
5	W 12×53	W 16×67	W 14×53	W 14×53
6	W 14×68	W 12×53	W 14×68	W 14×68
7	W 10×54	W 14×74	W 10×54	W 12×53
8	W 12×53	W 10×60	W 12×53	W 12×50
9	W 24×55	W 21×44	W 21×44	W 21×44
10	W 18×46	W 21×44	W 18×46	W 18×46
11	W 24×55	W 21×44	W 18×46	W 21×44
12	W 21×50	W 18×46	W 16×45	W 21×44
Weight (kN)	424.42	388.08	390.16	381.51
Top storey drift (cm)	3.45	4.24	4.81	4.74
Max. inter-storey drift (cm)	*	*	0.64	0.63
Max. interaction ratio	*	0.99	0.92	0.98
Number of structural analyses	28000	12560	6480	6960

15.85 cm. Size constraints presented in Eqs. (7) and (8) also are imposed on the design. The parameters used in this example are selected the same as the ones of the first example.

Table 5 compares the ABC algorithm result with the GA (Hayalioglu and Degertekin 2005) and the HS (Degertekin and Hayalioglu 2010) algorithms. Table 5 shows that the ABC algorithm yielded 10% and 1.7% lighter design than those of the GA (Hayalioglu and Degertekin 2005) and the HS (Degertekin and Hayalioglu 2010), respectively. For 20 runs, the average weight is obtained as 385.21 kN, with a standard deviation of 3.60 kN.

The highest interaction ratio for each group at the optimum design is displayed in Fig. 9. It is obvious from Fig. 9 that the highest interaction ratio for the seven member group is within 90% of maximum interaction ratio at the optimum. However, it is seen from Table 5 that top storey drift is far below the allowable sway of the top storey. This indicates that strength constraints are dominant at the optimum design.

The convergence curve for the ABC algorithm also is shown in Fig. 10. The ABC required 6960 structural analyses whereas the GA (Hayalioglu and Degertekin 2005) and the HS (Degertekin and

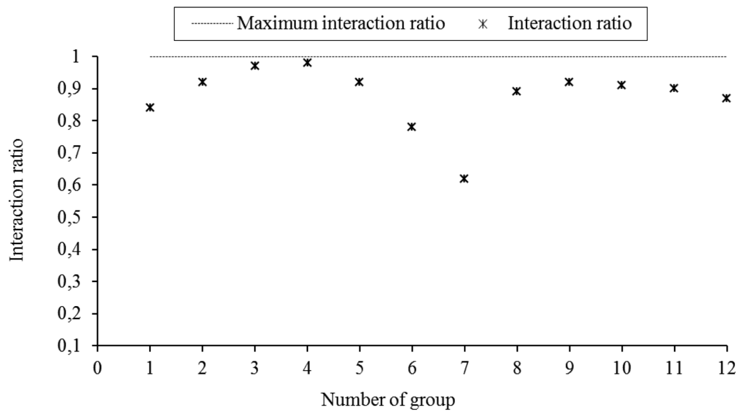

Fig. 9 The highest interaction ratio for each group for the four-bay, ten-storey frame

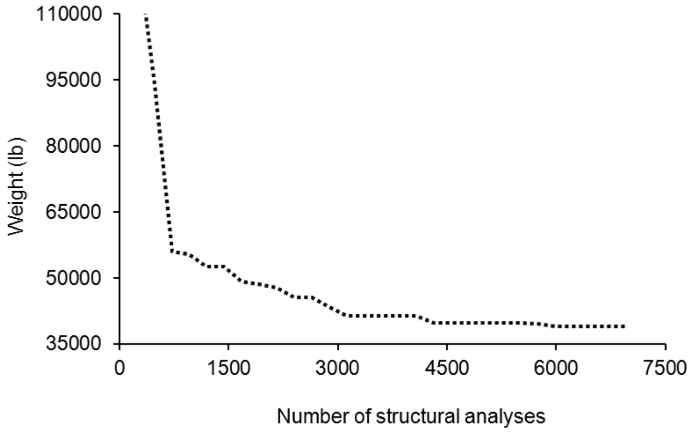

Fig. 10 Convergence curve for the four-bay, ten-storey frame

Hayalioglu 2010) converged to the optimum designs after 28000 and 12560 structural analyses, respectively.

6. Conclusions

This study presented an artificial bee colony algorithm for the optimum design of geometrically non-linear steel frames. The ABC algorithm was tested in three steel frames taken from literature. Optimization results suggest the following considerations. The ABC algorithm implemented in this study found better designs than the GA and the HS algorithms presented in literature. The computational effort of the ABC was lower than that observed for the GA and the HS algorithms.

It is important to note that standard deviation of optimized weights obtained over 20 independent runs always was quite small compared with average optimized weight. This indicates that the ABC algorithm presented in this paper possess the inherent ability of converging to a nearly global optimum design.

However, it should be noted that although the ABC found lighter designs than other meta-heuristic algorithms, it seemed that the ABC algorithm did not showed significantly computational advantage. In view of this, it appears that gradient-based ABC algorithm will be an important area for further research aimed at reducing the computational cost of the ABC algorithm.

References

AISC-LRFD (2001), "Manual of steel construction-Load and resistance factor design," American Institute of Steel Construction, Chicago.

Camp, C.V., Pezeshk, S. and Cao, G. (1998), "Optimized design of two-dimensional structures using a genetic algorithm," *J. Struct. Eng.-ASCE*, **124**(5), 551-559.

Camp, C.V., Bichon, B.J. and Stovall, P.S. (2005), "Design of steel frames using ant colony optimization," *J. Struct. Eng.- ASCE*, **131**(3), 369-379.

Deb, K. (2000), "An efficient constraint handling method for genetic algorithms," *Comput. Method Appl. M.*, **186**(2-4), 311-338.

Degertekin, S.O. (2008a), "Harmony search algorithm for optimum design of steel frame structures: a comparative study with other optimization methods," *Struct. Eng. Mech.*, **29**(4), 391-410.

Degertekin, S.O. (2008b), "Optimum design of steel frames using harmony search algorithm," *Struct. Multidiscip. O.*, **36**(4), 393-401.

Degertekin, S.O., Saka, M.P. and Hayalioglu, M.S. (2008a), "Optimal load and resistance factor design of geometrically nonlinear steel space frames via tabu search and genetic algorithm," *Eng. Struct.*, **30**(1), 197-205.

Degertekin, S.O., Hayalioglu, M.S. and Ulker, M. (2008b), "A hybrid tabu-simulated annealing heuristic algorithm for optimum design of steel frames," *Steel Compos. Struct.*, **8**(6), 475-490.

Degertekin, S.O. and Hayalioglu, M.S. (2010), "Harmony search algorithm for minimum cost design of steel frames with semi-rigid connections and column bases," *Struct. Multidiscip. O.*, **42**(5), 755-768.

Dorigo, M., Maniezzo, V. and Colorni, A. (1992), "An investigation of some properties of an ant algorithm," *Proc. 1992 Parallel Problem Solving from Nature Conf.*, Elsevier, Amsterdam.

Dumonteil, P., (1992), "Simple equations for effective length factors," *Eng. J. AISC*, **29**(3), 111-115.

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

Hasancebi, O., Carbas, S., Dogan, E., Erdal, F. and Saka, M.P. (2009), "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures," *Comput. Struct.*, **87**(5-6), 284-302.

Hasancebi, O., Carbas, S., Dogan, E., Erdal, F. and Saka, M.P. (2010), "Comparison of non-deterministic search techniques in the optimum design of real size steel frames," *Comput. Struct.*, **88**(17-18), 1033-1048.

Hayalioglu, M.S. and Degertekin, S.O. (2005), "Minimum cost design of steel frames with semi-rigid connections and column bases via genetic optimization," *Comput. Struct.*, **83**(21-22), 1849-1863.

Hemamalini, S. and Simon, S.P. (2010), "Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions," *Electr. Pow. Compo. Sys.*, **38**(7), 786-803.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

Karaboga, D. (2005), "An idea based on honey bee swarm for numerical optimization," Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

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

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

Karaboga, N. (2009), "A new design method based on artificial bee colony algorithm for digital IIR filters," *J. Frankl. Inst.*, **346**(4), 328-348.

Karaboga, D. and Akay, B. (2009), "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, **214**(1), 108-132.

Karaboga, D. and Akay, B. (2011), "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," *Appl. Soft Comput* .**11**(3), 3021-3031.

Kaveh, A. and Shojaee, S. (2007), "Optimal design of skeletal structures using ant colony optimization", *Int. J. Numer. Meth. Eng.*, **70**(5), 563-581.

Kennedy, J. and Eberhart, R.C. (1995), "Particle Swarm Optimization", *IEEE International Conference on Neural Networks*, 1942-1948.

Lamberti, L. and Pappalettere, C. (2009), "An improved harmony-search algorithm for truss structure optimization," *Proceedings of the Twelfth International Conference Civil, Structural and Environmental Engineering Computing*, Eds. Topping BHV, Neves LFC, Barros, RC Civil-Comp Press, Stirlingshire, Scotland.

Liu, M., Burns, S.A. and Wen, Y.K. (2006), "Genetic algorithm based construction-conscious minimum weight design of seismic steel moment-resisting frames," *J. Struct. Eng.- ASCE*, **132**(1), 50-58.

Omkar, S.N., Senthilnath, J., Khandelwal, R., Nail, G.N. and Gopalakrishnan, S. (2011), "Artificial bee colony (ABC) for multi-objective design optimization," *Appl. Soft Comput.*, **11**(1), 489-499.

Pezeshk, S., Camp, C.V. and Chen, D. (2000), "Design of nonlinear framed structures using genetic optimization," *J. Struct. Eng.- ASCE*, **126**(3), 382-388.

Saka, M.P. (2009). "Optimum design of steel sway frames to BS 5950 using harmony search algorithm," *J. Constr. Steel Res.*, **65**(1), 36-43.

Singh, A. (2009), "An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem," *Appl. Soft Comput.*, **9**(2), 625-631.

Sonmez, M. (2011a), "Discrete optimum design of truss structures using artificial bee colony algorithm," *Struct. Multidiscip. O.*, **43**(1), 85-97.

Sonmez, M. (2011b), "Artificial bee colony algorithm for optimization of truss optimization," *Appl. Soft Comput.*, **11**(2), 2406-2418.

Toropov, V.V. and Mahfouz, S.Y. (2001), "Design optimization of structural steelwork using a genetic algorithm, FEM and a system of design rules," *Eng. Comput.*, **18**(3-4), 437-459.

CC