OPTIMIZATION OF SKELETAL STRUCTURAL USING ARTIFICIAL BEE COLONY ALGORITHM

S. Talatahari*,†, M. Nouri‡ and F. Tadbiri‡

*Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran
‡Department of Structural Engineering, Shabestar Branch, Islamic Azad University-Shabestar, Iran

ABSTRACT

Over the past few years, swarm intelligence based optimization techniques such as ant colony optimization and particle swarm optimization have received considerable attention from engineering researchers. These algorithms have been used in the solution of various structural optimization problems where the main goal is to minimize the weight of structures while satisfying all design requirements imposed by design codes. In this paper, artificial bee colony algorithm (ABC) is utilized to optimize different skeletal structures. The results of the ABC are compared with the results of other optimization algorithms from the literature to show the efficiency of this technique for structural design problems.

Received: 20 June 2012; Accepted: 30 September 2012

KEYWORD: artificial bee colony algorithm; structural design problems; skeletal structures; trusses; frames.

1. INTRODUCTION

Optimization can be defined as finding solution of problems where it is necessary to maximize or minimize an objective function within a search domain which contains the acceptable values of variables while some restrictions are to be satisfied. There might be the large number of variables in the search domain that maximizes or minimizes the objective function while satisfying the restrictions. They are called feasible solutions and the solution
In structural optimization problems, the main goal is to minimize the weight or cost of structures while satisfying necessary limitation of structure. To achieve this goal, over the last decades many elegant and artificial optimization techniques have been successfully applied to a wide range of structural optimization problems. In recent years, direct search techniques based on the models of social interaction amongst organisms have been found capable of producing very powerful and robust search mechanisms [1]. Techniques belonging to this field imitate the collective behavior of a group of social insects (bees, termites, ants and wasps) to solve complex optimization problems. These insects live together in a nest and divide up the work tasks such as foraging, nest building and defense within the colony. Members of the colony perform their tasks by interacting or communicating in a direct or indirect manner in their local environment. The key feature of colony behavior is that even if one or some individuals fail in carrying out their task, the group as whole can still perform their tasks [2]. From the collective behavior in certain insect species emerges the swarm intelligence. Swarm intelligence based algorithms including particle swarm optimization (PSO) [3] and ant colony optimization (ACO) algorithms [4] have already been used for the weight minimization of structures involving discrete and continuous design variables. Swarm-based algorithms simulating bee swarm intelligence recently found new application areas in engineering design optimization. Karaboga and Basturk [5] presented a survey of these algorithms and their applications. Most of these algorithms make use of the bee metaphor imitating the food foraging behaviors of honeybees such as, the bee colony optimization algorithm [6,7], the Virtual Bee algorithm [8], the bee algorithm [9,10] and the Artificial bee colony algorithm [11-13].

Karaboga and Basturk [11,12] proposed the artificial bee colony (ABC) algorithm for unconstrained and constrained function optimization problems. The performance of the ABC algorithm was compared to that of differential evaluation, particle swarm optimization and an evolutionary algorithm. They declared that the ABC algorithm performed better than these methods and it can be effectively employed to solve engineering problems [14,15].

There are many research applied the ABC algorithm to different optimization algorithms. This study utilizes the ABC Algorithm to optimum design of skeletal structures. The rest of the paper is organized as follows: Section 2 presents the formulation of optimum design of structures. The framework of the ABC algorithm is described in Section 3. Numerical examples are presented in Section 4 and finally Section 5 concludes the paper.

2. OPTIMUM DESIGN OF STRUCTURE

The objective of optimization is to find a set of design variables that has the minimum weight and also satisfies certain constraints, as

\[
\text{Find } \{x\} = \{x_1, x_2, \ldots, x_n\} \quad \text{subject to } x_i \in D_i
\]
OPTIMIZATION OF SKELETAL STRUCTURAL USING ARTIFICIAL BEE...  

To minimize \[ w(x) = \sum_{i=1}^{nm} \phi_i d_i x_i \]  
subject to \[ g_j(x) \leq 0, j = 1, 2, ..., n \]

where \( \{x\} \) is the set of design variables; \( ng \) is the number of design variables; \( D_i \) is the allowable set of values for the design variable \( x_i \); \( w(\{x\}) \) presents the weight of the structure; \( nm \) is the number of members of the structure; \( \phi_i \) denotes the material density of member \( i \); \( L_i \) and \( x_i \) are the length and the cross-sectional of member \( i \), respectively; \( g_j(\{x\}) \) denotes design constraints; and \( n \) is the number of the constraints.

\( D_i \) can be considered either as a continuous set or as a discrete one [16]. In the continuous problems, the design variables can vary continuously in the optimization process

\[ D_i = \{ x_i | x_i \in [x_{i,\text{min}}, x_{i,\text{max}}] \} \]

where \( x_{i,\text{min}} \) and \( x_{i,\text{max}} \) are minimum and maximum allowable values for the design variable \( x_i \), respectively. If the design variables represent a selection from a set of parts as

\[ D_i = \{ d_{i,1}, d_{i,2}, ..., d_{i,r} \} \]

then the problem is considered as a discrete one, where \( r \) is the number of available discrete values. In order to handle the constraints, a penalty approach is utilized. In this method, the aim of the optimization is redefined by introducing the penalty function as

\[ f_{\text{penalty}}(x) = (1 + \varepsilon_1 v)^\varepsilon_2 \times w(\{x\}) \quad v = \sum_{i=1}^{n} \max(0, g_{i,\text{max}}) \]

where \( V \) denotes the sum of the violations of the design. The constant \( \varepsilon_1 \) and \( \varepsilon_2 \) are selected considering the exploration and the exploitation rate of the search space. Here, \( \varepsilon_1 \) is set to \( 0.9 \), \( \varepsilon_2 \) is set unity [17].

This paper investigates two types of skeletal structures consisting of trusses and frames. The constraint conditions for these structures are briefly explained in the following subsections.

2.1. Constraint conditions for truss structures

For truss structures, the stress limitations of the members are imposed according to the provisions of ASD-AISC [18] as follows:

\[
\begin{cases} 
\sigma_i^- = 0.6 f_y \\
\sigma_i^+ = 0.0 \quad \sigma_i \geq 0 \\
\sigma_i^- = 0.0 \quad \sigma_i < 0 
\end{cases}
\]

where \( \sigma_i^- \) is calculated according to the slenderness ratio:
where $E$ is the modulus of elasticity; $F_y$ is the yield stress of steel; $C_c$ denotes the slenderness ratio dividing the elastic and inelastic buckling regions; $\lambda$ presents the slenderness ratio.

The other constraint is the limitation of the nodal displacements, as

$$\delta_i \leq \delta_i^u \quad i = 1, 2, \ldots, nn$$

(9)

where $\delta_i$ is the nodal deflection; $\delta_i^u$ is the allowable deflection of node $i$; and $nn$ is the number of nodes.

2.2. Constraint conditions for steel frames

Optimal design of frame structures is subjected to the following constrains according to LRFD-AISC [19] provisions:

$$\frac{\Delta T}{H} \leq R$$

(10)

$$\frac{d_i}{h_i} \leq R_i \quad i = 1, 2, \ldots, ns$$

(11)

If

$$\frac{p_r}{\varphi_i p_n} \geq 0.2 \quad \frac{p_r}{\varphi_i p_n} + \frac{8}{9} \left(\frac{M_{rx}}{\varphi_b M_{nx}} + \frac{M_{ry}}{\varphi_b M_{ny}}\right) \leq 1.0$$

(12)

If

$$\frac{p_r}{\varphi_i p_n} < 0.2 \quad \frac{p_r}{2\varphi_i p_n} + \left(\frac{M_{rx}}{\varphi_b M_{nx}} + \frac{M_{ry}}{\varphi_b M_{ny}}\right) \leq 1.0$$

(13)

where $\Delta_T$ is the maximum lateral displacement; $H$ is the height of the frame structure; $R$ is the maximum drift index (1/300); $d_i$ is the inter-story drift; $h_i$ is the story height of the $i$th floor; $ns$ is the total number of stories; $R_i$ presents the inter-story drift index permitted by the code of the practice (1/300); $P_n$ is the required strength (tension or compression); $P_n$ is the nominal axial strength (tension or compression); $\varphi_i$ is the resistance factor ( $\varphi_i = 0.9$ for tension, $\varphi_i = 0.85$ for compression); $M_{rx}$ and $M_{ry}$ are the required flexural strengths in the $x$ and $y$ directions, respectively; $M_{nx}$ and $M_{ny}$ are the nominal flexural strengths in the $x$ and $y$ directions (for two-dimensional structures, $M_{ny} = 0$); and $\varphi_b$ denotes the flexural resistance reduction factor ($\varphi_b = 0.90$).
3. ARTIFICIAL BEE COLONY ALGORITHM

The foraging behavior, learning, memorizing and information sharing characteristics of honeybees have recently been one of the most interesting research areas in swarm intelligence. The ABC algorithm as a swarm intelligent optimization algorithm is inspired by honey bee foraging. This section reviews the framework of the algorithm, briefly.

3.1. General aspects

The ABC provides a population based search procedure in which individuals called foods positions are modified by the artificial bees with time and the bees aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. In the ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one. Thus, the ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process. This model that leads to the emergence of collective intelligence of honeybee swarms consists of three essential components: food sources, employed foragers, and unemployed foragers, and defines two leading modes of the honeybee colony behavior: requirement to a food source and abandonment of a source. The main components of this model are as below:

1. **Food sources:** In order to select a food source, a forager bee evaluates several properties related with the food source such as its closeness to the hive, richness of the energy, taste of its nectar, and the ease or difficulty of extracting this energy. For the simplicity, the quality of a food source can be represented by only one quantity although it depends on various parameters mentioned above.

2. **Employed foragers:** An employed forager carries information about her specific source and shares it with other bees waiting in the hive. The information includes the distance, the direction and the profitability of the food source.

3. **Unemployed foragers:** A forager bee that looks for a food source to exploit is called unemployed. It can be either a scout who searches the environment randomly or an onlooker who tries to find a food source by means of the information given by the employed bee.

3.2. The algorithm

The flowchart of the ABC algorithm is given in Figure 1. Each cycle of the search consists of three steps after initialization stage: placing the employed bees onto the food sources and calculating their nectar amounts; placing the onlookers onto the food sources and calculating the nectar amounts; and determining the scout bees and placing them onto the randomly determined food sources.
In the ABC algorithm, the first half of the colony consists of the employed artificial bees and the second half includes the onlookers. In this algorithm, for every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been abandoned becomes a scout.

The position of a food source represents a possible solution to the considered optimization problem and the nectar amount of the food source corresponds to the quality or fitness of the associated solution. The number of the employed bees or onlooker bees is equal to the number of solutions in the population. In the first step, the ABC algorithm generates randomly distributed predefined number of initial population, \( P \in SN \). Each position of the food source, \( x_{ijk} \) is three-dimensional in nature with \( i=1,2,...,SN; j=1,2,...,D \) and \( k=1,2,...,V \); where \( D \) is the dimension of each variable and \( V \) is the number of variables in the objective function. After initialization, the population of the positions (solutions) is subjected to repeated cycles,
C=1,2,…,MCN (maximum cycle number) of the search process of the employed bees, onlooker bees and scout bees. An employed bee produces a modification on the solution in its memory depending on the local information and tests the nectar amount (fitness value) of the new food source (new solution). Provided that the nectar amount of the new source is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise, it keeps the position of the previous source in its memory. When all the employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlooker bees in the dance area. An onlooker bee evaluates the nectar information taken from all the employed bees and selects a food source with a probability related to its nectar amount. As in the case of an employed bee, the onlooker bee produces a modification on the position in its memory and checks the nectar amount of the candidate source. If its nectar amount is higher than that of the previous one, the onlooker bee memorizes the new position and forgets the old one.

4. NUMERICAL EXAMPLES

Four truss and frame examples are selected to show efficiency and validation of the ABC algorithm containing:

- A 25-bar truss
- A 72-bar truss
- A 1-bay, 8 story frame
- A 3-bay, 15-story frame

All of the structural analysis as well as the optimization process are performed by MATLAB software and the ABC algorithm parameters are set as follows: a colony of bees size NP=50, the maximum number of cycle MNC=300, and LIMIT=50.

4.1. A 25-bar truss

The topology and nodal numbers of a 25-bar spatial truss structure are shown in Figure 2. In this example, designs for a multiple load case are performed and the results are compared to those of other optimization techniques [14,20,21]. The material density is considered as 0.1 lb/in$^3$ (2767.990 kg/m$^3$) and the modulus of elasticity is taken as 10,000 ksi (68,950 MPa). This spatial truss is subjected to two loading as described in [20].

In order to study the effect of the colony size on the convergence rate of the ABC algorithm, ten different colonies consisting of 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 bees were used. The averages of each set of 10 independent runs for each colony are given in Figure 3 where the objective function versus cycle numbers is shown. It can be seen from this figure that the convergence rates increase with greater numbers of bees. After 300 cycles, with the exception of the colony of 10 bees the results of all the colonies are very close to each other and they have almost the same weight. The colony size may be set at any value between 30 and 100. However, larger values increase the required analysis number and with small ones, the probability of losing the optimum design increase; as a result, in the current research the colony size is set at 50 bees for all examples. Table 1 compares the obtained result by this study with other methods. The present algorithm needs only 15000
analyses to find the optimum result while it is 30000 for standard ABC [14].

Figure 2. Twenty five bar space truss

Figure 3. Comparison of the convergence rates of ABC with ten different colony sizes

Table 1. Performance comparison for 25-bar space truss
---|---|---|---|---|---
1 | A₁ | 0.01 | 0.01 | 0.011 | 0.01
2 | A₂ - A₅ | 2.042 | 2.092 | 1.979 | 2.1157
3 | A₆ - A₉ | 3.001 | 2.964 | 3.003 | 2.9149
4 | A₁₀ - A₁₁ | 0.01 | 0.01 | 0.01 | 0.01
5 | A₁₂ - A₁₃ | 0.01 | 0.01 | 0.01 | 0.01
6 | A₁₄ - A₁₇ | 0.684 | 0.689 | 0.690 | 0.7832
7 | A₁₈ - A₂₁ | 1.625 | 1.601 | 1.679 | 1.6032
8 | A₂₂ - A₂₅ | 2.762 | 2.686 | 2.652 | 2.5654

| Best weight | 545.03 | 545.38 | 545.20 | 545.20 |
| Analysis number | 3502 | 20566 | 30000 | 15000 |

3.2. 72-bar truss

For the 72-bar space truss structure shown in Figure 4, the material density is 0.1 lb/in³ (2767.990 kg/m³) and the modulus of elasticity is 10,000 ksi (68,950 MPa). The members are subjected to the stress limits of 25 ksi (172.375 MPa). The uppermost nodes are subjected to the displacement limits of 0.25 in (0.635 cm) in both the x and y directions. The minimum permitted cross-sectional area of each member is 0.10 in² (0.6452 cm²), and the maximum cross-sectional area of each member is 4.00 in² (25.81 cm²).

![Figure 4. 72-bar spatial truss](image_url)

Table 2. Performance comparison for 72-bar spatial truss
## Table 1. Optimal Cross Section Area

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 $A_1 - A_4$</td>
<td>1.948</td>
<td>1.8577</td>
<td>1.755</td>
<td>1.909</td>
</tr>
<tr>
<td>2 $A_5 - A_{12}$</td>
<td>0.508</td>
<td>0.5059</td>
<td>0.505</td>
<td>0.520</td>
</tr>
<tr>
<td>3 $A_{13} - A_{16}$</td>
<td>0.101</td>
<td>0.1000</td>
<td>0.105</td>
<td>0.100</td>
</tr>
<tr>
<td>4 $A_{17} - A_{18}$</td>
<td>0.102</td>
<td>0.1000</td>
<td>0.155</td>
<td>0.100</td>
</tr>
<tr>
<td>5 $A_{19} - A_{22}$</td>
<td>1.303</td>
<td>1.2476</td>
<td>1.155</td>
<td>1.284</td>
</tr>
<tr>
<td>6 $A_{23} - A_{30}$</td>
<td>0.511</td>
<td>0.5269</td>
<td>0.585</td>
<td>0.503</td>
</tr>
<tr>
<td>7 $A_{31} - A_{34}$</td>
<td>0.101</td>
<td>0.1000</td>
<td>0.100</td>
<td>0.100</td>
</tr>
<tr>
<td>8 $A_{33} - A_{36}$</td>
<td>0.107</td>
<td>0.1012</td>
<td>0.100</td>
<td>0.100</td>
</tr>
<tr>
<td>9 $A_{37} - A_{40}$</td>
<td>0.561</td>
<td>0.5209</td>
<td>0.460</td>
<td>0.512</td>
</tr>
<tr>
<td>10 $A_{41} - A_{48}$</td>
<td>0.492</td>
<td>0.5172</td>
<td>0.530</td>
<td>0.523</td>
</tr>
<tr>
<td>11 $A_{49} - A_{52}$</td>
<td>0.100</td>
<td>0.1004</td>
<td>0.120</td>
<td>0.100</td>
</tr>
<tr>
<td>12 $A_{53} - A_{54}$</td>
<td>0.117</td>
<td>0.1005</td>
<td>0.165</td>
<td>0.100</td>
</tr>
<tr>
<td>13 $A_{55} - A_{58}$</td>
<td>0.156</td>
<td>0.1565</td>
<td>0.155</td>
<td>0.157</td>
</tr>
<tr>
<td>14 $A_{59} - A_{66}$</td>
<td>0.550</td>
<td>0.5507</td>
<td>0.535</td>
<td>0.537</td>
</tr>
<tr>
<td>15 $A_{67} - A_{70}$</td>
<td>0.390</td>
<td>0.3922</td>
<td>0.480</td>
<td>0.410</td>
</tr>
<tr>
<td>16 $A_{71} - A_{72}$</td>
<td>0.592</td>
<td>0.5922</td>
<td>0.520</td>
<td>0.563</td>
</tr>
<tr>
<td><strong>Best weight</strong></td>
<td>380.24</td>
<td>379.85</td>
<td>385.76</td>
<td>379.70</td>
</tr>
<tr>
<td><strong>Analysis number</strong></td>
<td>18500</td>
<td>20566</td>
<td>---</td>
<td>15000</td>
</tr>
</tbody>
</table>

**Figure 5. Convergence history of the best result obtained by the ABC**

Figure 5 presents the convergence characteristic curve of the ABC. For this spatial truss structure, it takes about 500 iterations (15000 analyses) for the ABC to converge, while it is
18500 and 20566 for the ACO [24] and BB-BC [21], respectively. Compared to the other methods, the best weight is belonged to the ABC with the weight of 379.70 lb. These results demonstrate the efficiency of the ABC compared to GA [23], ACO and BB-BC methods. Table 2 compares the performance of the ABC algorithm with those of the previously reported algorithms in the literature.

3.3. 1-bay 8-story frame

Figure 6 shows the configuration of the 1-bay 8-story framed structure and applied loads. Several researchers have developed design procedures for this frame; Camp et al. [25] used a genetic algorithm, Kaveh and Shojaee [26] utilized ACO and Kaveh and Talatahari [22] applied an improved ACO to solve this problem.

The ABC algorithm found the optimal weight of the one-bay eight-story frame to be 30.91 kN which is the best one compared to the other method. Table 3 lists the optimal
values of the eight design variables obtained by this research, and compares them with other results.

Table 3. Optimal design comparison for the 1-bay 8-story frame

<table>
<thead>
<tr>
<th>Element group</th>
<th>Optimal W-shaped sections</th>
<th>GA [25]</th>
<th>ACO [26]</th>
<th>IACO [22]</th>
<th>This study</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>W18X35 W16X26 W21X44</td>
<td></td>
<td></td>
<td></td>
<td>W21X44</td>
</tr>
<tr>
<td>2</td>
<td>W18X35 W18X40 W18X35</td>
<td></td>
<td></td>
<td></td>
<td>W16X26</td>
</tr>
<tr>
<td>3</td>
<td>W18X35 W18X35 W18X35</td>
<td></td>
<td></td>
<td></td>
<td>W14X22</td>
</tr>
<tr>
<td>4</td>
<td>W18X26 W14X22 W12X22</td>
<td></td>
<td></td>
<td></td>
<td>W12X16</td>
</tr>
<tr>
<td>5</td>
<td>W18X46 W21X50 W18X40</td>
<td></td>
<td></td>
<td></td>
<td>W18X35</td>
</tr>
<tr>
<td>6</td>
<td>W16X31 W16X26 W16X26</td>
<td></td>
<td></td>
<td></td>
<td>W18X35</td>
</tr>
<tr>
<td>7</td>
<td>W16X26 W16X26 W16X26</td>
<td></td>
<td></td>
<td></td>
<td>W18X35</td>
</tr>
<tr>
<td>8</td>
<td>W12X16 W12X14 W12X14</td>
<td></td>
<td></td>
<td></td>
<td>W16X26</td>
</tr>
<tr>
<td>Weight (kN)</td>
<td>32.83 31.68 31.05 30.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.4. Design of a 3-bay, 15-story frame

The configuration and applied loads of a 3-bay 15-story frame structure is shown in Figure 7. The displacement and AISC combined strength constraints are the performance constraint of this frame.

Table 4. Optimal design comparison for the 3-bay, 15-story frame

<table>
<thead>
<tr>
<th>Element group</th>
<th>Optimal W-shaped sections</th>
<th>PSO [16]</th>
<th>HBB-BC [28]</th>
<th>ICA [27]</th>
<th>This study</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>W33X118 W24X117 W24X117</td>
<td>W36X135</td>
<td>W27X146</td>
<td>W18X93</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>W33X263 W21X132 W21X147</td>
<td>W27X84</td>
<td>W24X94</td>
<td>W21X93</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>W24X76 W12X95 W27X84</td>
<td>W14X74</td>
<td>W24X68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>W36X256 W18X119 W27X114</td>
<td>W14X109</td>
<td>W21X44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>W21X73 W21X93 W14X74</td>
<td>W18X86</td>
<td>W21X44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>W18X86 W18X97 W18X86</td>
<td>W18X93</td>
<td>W21X44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>W18X65 W18X76 W12X96</td>
<td>W10X39</td>
<td>W16X36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>W21X68 W18X65 W24X68</td>
<td>W18X65</td>
<td>W16X40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>W18X60 W18X60 W10X39</td>
<td>W16X36</td>
<td>W21X44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>W18X65 W10X39 W12X40</td>
<td>W16X40</td>
<td>W21X44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>W21X44 W21X48 W21X44</td>
<td>W21X44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight (kips)</td>
<td>111.66 97.69 93.85 93.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The optimum design of the frame obtained by using ABC has the minimum weight of 93.76 kip. The optimum designs for PSO [16], HBB-BC [28] and ICA [27] had the weight...
of 111.66 kips, 97.69 kips and 93.85 kips, respectively. Table 4 summarizes the optimal results for the various algorithms. Clearly, it can be seen that the present algorithm can find better design.

5. CONCLUSION

The ABC algorithm, based on mimicking the food foraging behavior of honeybee swarms,
is proposed to solve structural optimization problems containing truss and frame structures. Optimization software based on the ABC algorithm was coded in the MATLAB with using object-oriented technology. Four test problems were studied using the optimization program to show the efficiency of the ABC algorithm. The comparison of the results of the ABC with those of other algorithms demonstrated that the ABC algorithm provides results as good as or better than other algorithms and can be used effectively for solving such problems. Expanding and hybridizing this method can provide a fruitful era to find more efficient and powerful methods to optimum design of skeletal structures.

REFERENCES