A Comparative Analysis of Selection Schemes in the Artificial Bee Colony Algorithm

Ajit Kumar¹, Dharmender Kumar², S.K. Jarial¹

¹ University of Science and Technology, Deenbandhu Chhotu Ram, Murthal, Sonepat, India
² University of Science and Technology, Guru Jambheshwar, Hisar, Haryana, India

{ajit.hisar, dharmindia24}@gmail.com, s.jarial@rediffmail.com

Abstract. The Artificial Bee Colony (ABC) algorithm is a popular swarm based algorithm inspired by the intelligent foraging behavior of honey bees. In the past, many swarm intelligence based techniques were introduced and proved their effective performance in solving various optimization problems. The exploitation of food sources is performed by onlooker bees in accordance with a proportional selection scheme that can be further modified to avoid such shortcomings as population diversity and premature convergence. In this paper, different selection schemes, namely, tournament selection, truncation selection, disruptive selection, linear dynamic scaling, linear ranking, sigma truncation, and exponential ranking have been used to analyze the performance of the ABC algorithm by testing on standard benchmark functions. From the simulation results, the schemes other than the standard ABC prove their efficient performance.

Keywords. Swarm based algorithm, artificial bee colony, optimization, selection scheme.

1 Introduction

A number of complex tasks are systematically performed by honey bees; a good example of such tasks is collection and processing of nectar [1]. The effectiveness and simplicity of the whole process is due to the decentralized decision making approach of honey bee colonies [2]. Such swarm intelligence features as autonomy, self-organizing, distributed functioning employed by a bee swarm provided inspiration to solve complex traffic, transportation problems [3, 4] and deterministic combinatorial problems in dynamic and uncertain environments [5, 6, 7]. Swarm intelligence algorithms based on the behavior of bees can be classified into two categories: the foraging behavior and the marriage behavior. Algorithms in the first category are inspired by searching for food sources and nest sites, while those of the second category are based on the marriage behavior [8]. One of the most important algorithms inspired by the foraging behavior of honey bee swarms is the Artificial Bee Colony (ABC). It was proposed by Karaboga and is used for solving various optimization problems [9, 10].

The remainder of the paper is organized as follows. Section 2 presents the original ABC algorithm and its selection scheme. Various selection schemes applied to the ABC are described in Section 3. The experimental results are presented and analyzed in Section 4. The paper is concluded in Section 5.

2 Artificial Bee Colony Algorithm

The ABC is a population based optimization algorithm which is iterative in nature. Basically, the ABC consists of cycles of four phases: the initialization phase, the employed bees phase, the onlooker bees phase, and the scout bees phase. The bees going to a food source already visited by them are the employed bees, while the bees looking for a food source are unemployed. The scout bees carry out search for new food sources, and the onlooker bees wait for the information from the employed bees for food sources. The information exchange among bees takes place...
through the waggle dance. There is one employed bee for every food source. An employed bee becomes scout when the position of a food source does not get improved through the predetermined number of attempts called “limit”. In this way, the exploitation process is performed by the employed and onlooker bees, whereas the scouts perform exploration of the search space [10].

There are three control parameters used in the ABC algorithm: the number of employed or onlooker bees to represent the number of food sources (N), the value of limit, the maximum cycle number (MCN). The main steps of the ABC are as follows.

- **Step 1.** Generate the initial population of solutions \( x_i^j \), \( i=1...N \), \( j=1...D \) using (1) and evaluate the fitness using (2).
- **Step 2.** Generate new solutions for the employed bees using (3) and evaluate the fitness.
- **Step 3.** Apply the greedy selection process for the employed bees.
- **Step 4.** Calculate the probability values for the current solution using (4) so that the onlooker bee can choose one according to its value.
- **Step 5.** Assign the onlooker bees to the solutions according to the probability, generate new solutions using (3) and evaluate the fitness.
- **Step 6.** Apply the greedy selection process for the onlooker bees.
- **Step 7.** If there is a solution abandoned by the bees, stop its exploitation and replace it with a new solution produced by (1).
- **Step 8.** Memorize the best solution found so far.
- **Step 9.** Check the termination criteria. If not satisfied, go to Step 2, otherwise end.

\[
x_i^j = x_{\text{min}}^j + \text{rand}(0,1)(x_{\text{max}}^j - x_{\text{min}}^j).
\]

where \( x_i^j \) is a parameter for the \( i \)th employed bee on the \( j \)th dimension, \( x_{\text{max}}^j \) and \( x_{\text{min}}^j \) are the upper and lower bounds for \( x_i^j \).

\[
fit_i = \begin{cases} 
\frac{1}{1 + f_i} & f_i \geq 0 \\
1 + \text{abs}(f_i) & f_i < 0 
\end{cases}
\]

where \( f_i \) is a specific objection function and \( fit_i \) is a fitness value.

\[
v_{ij} = x_{ij} + \phi (x_{ij} - x_{kj}),
\]

where \( i, k \in \{1...N\}, i \neq k \) and \( j \in \{1...D\}, x_{ij} \) is the \( i \)th employed bee in the \( j \)th dimension, \( v_{ij} \) is a new solution for \( x_{ij} \), \( x_{kj} \) is the neighbor of \( x_{ij} \), \( \phi \) is a random number in the range [-1,1] to control the production of neighbor solutions around \( x_{ij} \).

\[
p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j},
\]

where \( fit_i \) is the fitness value of the \( i \)th solution and \( p_i \) is the selection probability of the \( i \)th solution.

### 2.1 Selection Scheme in the Basic ABC

As explained above, food sources are chosen by the onlooker bees using a stochastic selection scheme in accordance with the probability value \( p_i \). The process employs three stages [11]:

(i) Calculate the fitness value using (2).
(ii) Calculate the probability value using (4).
(iii) Choose a food source according to the probability value based on the roulette wheel method.

However, the proportional selection scheme employed in the ABC has two shortcomings viz. reduction in population diversity and premature convergence. Thus, the ABC is not able to maintain the balance between exploration (diversification) and exploitation (intensification) of the search space and is considered as an inefficient algorithm.

### 3 Description of Selection Schemes

The selection scheme plays an important role in the ABC algorithm as it drives the search space in a proper direction. These schemes may be classified in two categories: proportionate selection and ordinal based selection. In the proportionate selection scheme, individuals are selected on the basis of their fitness values relative to the fitness of others, whereas in the ordinal based scheme, individuals are selected based on their rank in the
population. The rank is determined in accordance with their fitness values. The schemes presented in this paper except the proportional selection in the basic ABC are covered in the ordinal based selection category. In this work, we performed experiments on the ABC using different selection schemes. The details of the schemes are given in what follows.

### 3.1 Tournament Selection

This selection scheme works by holding a tournament of \( N \) individuals chosen from the population, where \( N \) is taken as the tournament size [11, 12, 13, 14]. The fitness values of individuals are compared and some score (say, \( s \)) is assigned to the best one. The process is repeated till the best in the population achieves the highest score. The individuals are then selected according to the probability using the following equation:

\[
P_i = \frac{S_i}{\sum_{i=1}^{N} S_i}.
\]  

### 3.2 Truncation Selection

This selection scheme assigns equal selection probabilities to the \( \mu \) best individuals selected in a population of size \( \lambda \) and is equivalent to \((\mu,\lambda)\)-selection used in evolution strategies [12, 15, 16]. The selection probabilities are given as

\[
P_i = \begin{cases} \frac{1}{\mu}, & 1 \leq i \leq \mu \\ 0, & \mu < i \leq \lambda. \end{cases}
\]  

### 3.3 Disruptive Selection

This scheme introduces the concept of normalized-by-mean fitness function. The idea is to give more chances to better and worse solutions in comparison to moderate solutions so that the population diversity can be improved [11, 17, 18]. The selection probability is calculated as follows:

\[
P_i = \frac{\text{fit}_i}{\sum_{i=1}^{N} \text{fit}_i}.
\]  

where \( \text{fit}_i \) is the fitness value of the \( i^{th} \) solution and \( P_i \) is the selection probability of the \( i^{th} \) solution. The fitness function is given by

\[
\text{fit}_i = |f_i - \bar{f}|,
\]  

where \( f_i \) is a specific objective function, \( \bar{f} \) is the average of the objective values for the individuals in the population.

### 3.4 Linear Dynamic Scaling

In order to improve the performance of the proportional selection, it is combined with a scaling technique called linear dynamic scaling [12]. The dynamic scaling is introduced to favor better individuals resulting in improved population fitness over generations. The selection probability is given by

\[
P_i = \frac{f_i - c}{S_f - \lambda c},
\]  

where \( S_f = \sum_{j=1}^{\lambda} f_j \), \( c > 0 \), and \( \lambda \) is the number of solutions in the population.

### 3.5 Linear Ranking

In this scheme, the ranks are assigned to the individuals based on their fitness values. The individual having the worst fitness is assigned rank 1 and the best fitness is assigned rank \( N \). The method uses a linear function to calculate selection probabilities according to the rank of individuals [12, 16]:

\[
P_i = \frac{1}{N} \left( \eta^+ + \frac{(\eta^+ - \eta^-)}{N - 1} \right), i \in \{1, \ldots, N\}.
\]  

To satisfy the constraints, two conditions must be fulfilled:

\[
\eta^+ = 2 - \eta^- \text{ and } \eta^- \geq 0.
\]

### 3.6 Sigma Truncation

In order to improve the fitness of a population, low fitness individuals are discarded using the standard deviation of fitness values before scaling.
them. This scheme ensures the selection of good fitness individuals [19, 20]. The fitness values of individuals are calculated as

$$f_{it}' = f_{it} - (\bar{f}_{it} - \sigma),$$

where $\bar{f}_{it}$ is the average fitness value of the population, $\sigma$ is the standard deviation of the fitness values, and $c$ is a small constant having values from 1 to 3.

### 3.7 Exponential Ranking

In this scheme, ranks are assigned to the individuals similar to linear ranking. The difference lies in exponential weighing of ranked individuals to compute probabilities as follows [12, 16]:

$$p_i = \frac{c - 1}{c^N - 1} c^{N-i}, i \in \{1, \ldots, N\},$$

where $c<1$, indicating the selection probability of the best individual.

### 4 Experimental Results and Discussions

#### 4.1 Test Problems

Six benchmark functions were used for simulation to evaluate the performance of various selection
### Table 2. Results of algorithms (varying maximum cycles)

[Colony Size=100, Limit=100, Parameters=100, Runs=10]

<table>
<thead>
<tr>
<th></th>
<th>MCN</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>10</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>250930</td>
<td>185124</td>
<td>128051</td>
<td>249890</td>
<td>172770</td>
<td>102611</td>
<td>247661</td>
<td>168817</td>
<td>104904</td>
<td>242485</td>
<td>128258</td>
<td>55800.9</td>
<td></td>
</tr>
<tr>
<td><strong>SD</strong></td>
<td>12025.3</td>
<td>10471.2</td>
<td>11303.3</td>
<td>12918.1</td>
<td>11439.3</td>
<td>13032.3</td>
<td>11757.9</td>
<td>12539</td>
<td>10245.3</td>
<td>15090.1</td>
<td>13879.3</td>
<td>16367.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DABC</th>
<th>TRABC</th>
<th>TABC</th>
<th>ABC</th>
<th>TRABC</th>
<th>TABC</th>
<th>ABC</th>
<th>DABC</th>
<th>TRABC</th>
<th>TABC</th>
<th>ABC</th>
<th>DABC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>3.559E+09</td>
<td>16367.7</td>
<td>125258</td>
<td>242465</td>
<td>128258</td>
<td>55800.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SD</strong></td>
<td>13879.3</td>
<td>16367.7</td>
<td>125258</td>
<td>242465</td>
<td>128258</td>
<td>55800.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Formulas

- **Sphere function:**
  \[ f_{1}(x) = \sum_{i=1}^{n} x_{i}^{2} - 100 \leq x_{i} \leq 100. \]  

- **Rosenbrock function:**
  \[ f_{2}(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_{i}^{2})^{2} + (x_{i} - 1)^{2}, \quad -100 \leq x_{i} \leq 100. \]  

- **Griewank function:**
  \[ f_{3}(x) = \sum_{i=1}^{n} \left( x_{i}^{2} \right) - 10 \cos(2\pi x_{i}) + 10, \quad \text{if} \quad 5.12 \leq x_{i} \leq 5.12. \]  

- **Ackley function:**
  \[ f_{4}(x) = -20 \exp\left(-\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_{i}^{2}}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_{i})\right) + 20 + e, \quad \text{if} \quad -600 \leq x_{i} \leq 600. \]
Table 3. Results of algorithms (varying colony size)

<table>
<thead>
<tr>
<th></th>
<th>ABC</th>
<th>TABC</th>
<th>TRABC</th>
<th>DABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>colony</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>155803</td>
<td>120075</td>
<td>121949</td>
<td>153139</td>
</tr>
<tr>
<td>50</td>
<td>114532</td>
<td>104069</td>
<td>111492</td>
<td>111246</td>
</tr>
<tr>
<td>100</td>
<td>98350.8</td>
<td>128334</td>
<td>64734</td>
<td>61148</td>
</tr>
<tr>
<td>Mean</td>
<td>12251.9</td>
<td>16413.2</td>
<td>8622.49</td>
<td>15228</td>
</tr>
<tr>
<td>SD</td>
<td>9963.29</td>
<td>40111.8</td>
<td>11485.9</td>
<td>9452.93</td>
</tr>
<tr>
<td>10</td>
<td>8.396E+10</td>
<td>5.61E+10</td>
<td>4.61E+10</td>
<td>6.355E+10</td>
</tr>
<tr>
<td>50</td>
<td>4.362E+10</td>
<td>4.281E+10</td>
<td>5.785E+10</td>
<td>5.765E+10</td>
</tr>
<tr>
<td>100</td>
<td>7.628E+10</td>
<td>7.787E+10</td>
<td>3.501E+10</td>
<td>6.086E+10</td>
</tr>
<tr>
<td>Mean</td>
<td>1.595E+10</td>
<td>7.42E+09</td>
<td>1.229E+10</td>
<td>1.575E+10</td>
</tr>
<tr>
<td>SD</td>
<td>6.642E+09</td>
<td>2.378E+10</td>
<td>6.267E+09</td>
<td>7.195E+09</td>
</tr>
<tr>
<td>10</td>
<td>1095.59</td>
<td>977.255</td>
<td>932.744</td>
<td>994.253</td>
</tr>
<tr>
<td>50</td>
<td>877.84</td>
<td>847.167</td>
<td>967.969</td>
<td>946.695</td>
</tr>
<tr>
<td>100</td>
<td>846.905</td>
<td>821.185</td>
<td>950.62</td>
<td>636.894</td>
</tr>
<tr>
<td>Mean</td>
<td>45.81</td>
<td>62.85</td>
<td>47.755</td>
<td>47.754</td>
</tr>
<tr>
<td>SD</td>
<td>53.519</td>
<td>52.881</td>
<td>77.5</td>
<td>47.25</td>
</tr>
<tr>
<td>10</td>
<td>1406.62</td>
<td>1137.28</td>
<td>969.686</td>
<td>1318.7</td>
</tr>
<tr>
<td>50</td>
<td>1066.59</td>
<td>930.254</td>
<td>1132.43</td>
<td>925.896</td>
</tr>
<tr>
<td>100</td>
<td>920.12</td>
<td>892.631</td>
<td>1189.65</td>
<td>616.013</td>
</tr>
<tr>
<td>Mean</td>
<td>142.168</td>
<td>120.243</td>
<td>134.866</td>
<td>126.881</td>
</tr>
<tr>
<td>SD</td>
<td>102.622</td>
<td>235.766</td>
<td>105.621</td>
<td>140.515</td>
</tr>
<tr>
<td>100</td>
<td>19.604</td>
<td>19.64</td>
<td>19.04</td>
<td>17.439</td>
</tr>
<tr>
<td>Mean</td>
<td>0.119</td>
<td>0.207</td>
<td>0.152</td>
<td>0.218</td>
</tr>
<tr>
<td>SD</td>
<td>0.294</td>
<td>0.215</td>
<td>0.615</td>
<td>0.223</td>
</tr>
<tr>
<td>10</td>
<td>-15851.9</td>
<td>-18777.1</td>
<td>-18969</td>
<td>-15924.3</td>
</tr>
<tr>
<td>50</td>
<td>-17914.9</td>
<td>-16753.7</td>
<td>-19305.4</td>
<td>-17754.3</td>
</tr>
<tr>
<td>100</td>
<td>-1137.28</td>
<td>-1026.66</td>
<td>-1089.4</td>
<td>-1137.28</td>
</tr>
<tr>
<td>Mean</td>
<td>-148.28</td>
<td>-106.99</td>
<td>-103.275</td>
<td>-110.568</td>
</tr>
<tr>
<td>SD</td>
<td>118.184</td>
<td>118.184</td>
<td>118.184</td>
<td>118.184</td>
</tr>
</tbody>
</table>

4.2 Experimental Settings

The algorithms for various selection schemes are implemented using MATLAB R2012a on an Intel (R) Core (TM) i3 CPU 3.06 GHZ with 4 GB RAM. In the following tables, ABC represents the original proportional scheme. TABC means the tournament selection, TRABC represents the truncation selection, DABC is the disruptive selection, LDABC
Table 4. Results of algorithms (varying initialization range)
(FR: Full Range, LHR: Left Half Range, RHR: Right Half Range)
[Colony size=100, Limit=100, Parameters=100, Max Cycles=100, Runs=10]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>SD</td>
<td>SD</td>
</tr>
<tr>
<td>TRABC</td>
<td>SD</td>
<td>SD</td>
</tr>
<tr>
<td>LDABC</td>
<td>SD</td>
<td>SD</td>
</tr>
<tr>
<td>STABC</td>
<td>SD</td>
<td>SD</td>
</tr>
<tr>
<td>ERABC</td>
<td>SD</td>
<td>SD</td>
</tr>
</tbody>
</table>

The experiments were performed on the six benchmark functions given above. In all the experiments, the limit was put to 1, and the maximum cycles, and limit were fixed as 100. The function values, the values of selection intensity, success rate, reproduction rate, and loss of diversity were also calculated.

4.3 Effect of Dimensions

We performed simulations on modified ABC algorithms to analyze the effect of varying dimensions of the problem. The colony size, maximum cycles, and limit were fixed as 100. The

Computación y Sistemas, Vol. 20, No. 1, 2016, pp. 55-66
doi: 10.13053/CyS-20-1-2228
performance of all ABC algorithms deteriorated as the dimension of the problem was increased (10, 50, 100).

The results in Table 1 show that STABC generated better results for Rastrigin and Ackley functions followed by LDABC for Sphere and Griewank functions in less dimensions, i.e., 10. Again, STABC produced excellent results with an increase in dimensions up to 50. However, DABC had superior performance for 100 dimensions. From Fig. 1(a), we can see that the increase in dimensions makes the convergence of DABC method better for Sphere function and also for Rastrigin function as given in Fig. 1(b).

### 4.4 Effect of Cycles

We analyzed the performance of the ABC algorithms by varying the maximum number of cycles. The experiment was repeated for the six benchmark functions as given in Table 2.

The obtained values prove better results for the sigma truncation scheme on Sphere, Rosenbrock, Rastrigin, and Griewank functions. Figs. 2(a) and 2(b) prove better results of STABC on Rosenbrock function and of DABC on Ackley function. For a less number of cycles, i.e. 10, LDABC shows the best performance.

### 4.5 Effect of Colony Size

In the next experiment, we determined what size of population is suitable to generate better results. The experiment was conducted for all six test problems. Table 3 presents better results in case of STABC on Rosenbrock, Griewank functions, and in case of DABC on Rastrigin, Ackley, Schwefel functions for varying colony sizes.

For a small colony size of 10, the results of TRABC are good on Sphere function. The performance of DABC got improved with an increase in the colony size as given in Figs. 3(a) and 3(b).

### 4.6 Effect of Region Scaling

We also investigated the effect of initializing the solutions in various sub-regions of the search space. There was a possibility of variation in the performance of the algorithms during initialization in the left half and the right half of the search space. The results of the experiments using different selection schemes are reported in Table 4. The aim is to determine the sensitivity of the algorithms in finding global optima under varying initialization ranges. All the ABC algorithms were found to be less sensitive to initial solutions in finding global optima as shown in Figs. 4(a) and 4(b).

Table 5. Results of algorithms

<table>
<thead>
<tr>
<th>ABC</th>
<th>TABC</th>
<th>TRABC</th>
<th>DABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>SR</td>
<td>RR</td>
<td>Pd</td>
</tr>
<tr>
<td>f1</td>
<td>0.058</td>
<td>100</td>
<td>1.062</td>
</tr>
<tr>
<td>f2</td>
<td>0.012</td>
<td>0</td>
<td>1.168</td>
</tr>
<tr>
<td>f3</td>
<td>0.009</td>
<td>81</td>
<td>1.052</td>
</tr>
<tr>
<td>f4</td>
<td>0.006</td>
<td>100</td>
<td>1.045</td>
</tr>
<tr>
<td>f5</td>
<td>0.015</td>
<td>100</td>
<td>1.041</td>
</tr>
<tr>
<td>f6</td>
<td>0.008</td>
<td>0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

LDABC | LRABC | STABC | ERABC

<table>
<thead>
<tr>
<th>SI</th>
<th>SR</th>
<th>RR</th>
<th>Pd</th>
<th>SI</th>
<th>SR</th>
<th>RR</th>
<th>Pd</th>
<th>SI</th>
<th>SR</th>
<th>RR</th>
<th>Pd</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>0.012</td>
<td>100</td>
<td>1.166</td>
<td>0.988</td>
<td>0.014</td>
<td>100</td>
<td>1.0</td>
<td>0.99</td>
<td>0.002</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td>f2</td>
<td>0.023</td>
<td>0</td>
<td>1.090</td>
<td>0.989</td>
<td>0.053</td>
<td>0</td>
<td>1.077</td>
<td>0.989</td>
<td>0.023</td>
<td>0</td>
<td>1.5</td>
</tr>
<tr>
<td>f3</td>
<td>0.029</td>
<td>74</td>
<td>1.043</td>
<td>0.989</td>
<td>0.018</td>
<td>11</td>
<td>1.045</td>
<td>0.989</td>
<td>0.001</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td>f4</td>
<td>0.028</td>
<td>100</td>
<td>1.0</td>
<td>0.99</td>
<td>0.073</td>
<td>100</td>
<td>1.031</td>
<td>0.989</td>
<td>0.016</td>
<td>100</td>
<td>1.042</td>
</tr>
<tr>
<td>f5</td>
<td>0.024</td>
<td>100</td>
<td>1.0</td>
<td>0.99</td>
<td>0.032</td>
<td>100</td>
<td>1.052</td>
<td>0.989</td>
<td>0.004</td>
<td>100</td>
<td>1.0</td>
</tr>
<tr>
<td>f6</td>
<td>0.006</td>
<td>0</td>
<td>1.0</td>
<td>0.99</td>
<td>0.008</td>
<td>0</td>
<td>1.0</td>
<td>0.99</td>
<td>0.001</td>
<td>0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4.6 Effect of Region Scaling

We also investigated the effect of initializing the solutions in various sub-regions of the search space. There was a possibility of variation in the performance of the algorithms during initialization in the left half and the right half of the search space. The results of the experiments using different selection schemes are reported in Table 4. The aim is to determine the sensitivity of the algorithms in finding global optima under varying initialization ranges. All the ABC algorithms were found to be less sensitive to initial solutions in finding global optima as shown in Figs. 4(a) and 4(b).
A Comparative Analysis of Selection Schemes in the Artificial Bee Colony Algorithm

Fig. 1(a). Sphere

Fig. 1(b). Rastrigin

Fig. 2(a). Rosenbrock

Fig. 2(b). Ackley

Fig. 3(a). Griewank

Fig. 3(b). Schwefel
4.7 Statistical Analysis

The proportional selection scheme used in the basic ABC lacks the driving force to attract better individuals which may result in premature convergence and a lack of population diversity. The tournament selection scheme randomly selects a number of N individuals and comparison is made based on their fitness values. The truncation selection scheme assigns equal selection probabilities to some selected best individuals in the population. The linear dynamic scaling scheme works by promoting better than average individuals at the cost of worse than average individuals. The linear ranking scheme is biased to favor the good fitness individuals in the population as the rank is assigned based on the fitness value. The exponential ranking scheme works in a similar manner to the linear ranking scheme except the use of the exponential function in computing selection probabilities.

From Figs. 1, 2, and 3, we can state that the DABC and STABC algorithms prove their effective performance in comparison to other algorithms. The disruptive selection scheme favors both high fitness and low fitness solutions and tends to maintain population diversity. Hence, this scheme improves the worse fitness solutions in concurrence with the high fitness solutions. In the case of STABC, the individuals having the fitness value less than c standard deviations of the average value are discarded, while a large portion of the population having the fitness values within c standard deviations of the average value are favored for selection.

Table 5 presents the analysis of the numerical results obtained with a slight change (i.e. 100 runs) in the experimental setting of subsection 4.2 using various selection schemes. Selection Intensity (SI) also called Selection Pressure measures the degree that drives the algorithm to improve the population fitness. It computes the difference between the population average fitness after and before selection. A high value of SI indicates high convergence rate, i.e. the algorithm is able to find optimal solutions early. Positive values of SI in Table 5 prove improvement in average fitness of the original ABC and the modified ABC algorithms due to selection for all test functions.

Success Rate (SR) shows that algorithm is able to obtain a desired function value (i.e. <2) using the given experimental settings. From the table, we can see that the success rate of the TRABC and STABC algorithms gets improved for Rastrigin function, whereas it is comparable to the original ABC for the remaining test functions.

Reproduction Rate (RR) is calculated to represent the ratio of the number of individuals with a certain fitness value after and before selection. A value of RR > 1 means better individuals are favored and bad individuals are discarded by a suitable selection scheme. Table 5 clearly shows that all selection schemes are able to replace bad individuals by better individuals.

Loss of Diversity (Pd) presents the ratio of the individuals of a population that are not selected during the selection stage. It means that Reproduction Rate and Loss of Diversity are
related to each other. The value of Pd should be as low as possible, as a high value of Pd may increase the risk of premature convergence. The values in the table clearly confirm the results.

5 Conclusions and Future Work

In this paper, we compared the performance of the Artificial Bee Colony algorithm combined with different selection schemes on six numerical optimization functions. The simulations were performed by varying the values of different control parameters used in the ABC algorithm in addition to initialization ranges. On the basis of the results obtained, an analysis is made in terms of selection intensity, success rate, reproduction rate, and loss of diversity.

With an increase in the number of dimensions, it becomes difficult to find optimal solutions in all selection schemes. As the number of cycles increases, the algorithms explore and exploit efficiently the search space to provide proper convergence and population diversity. An increase in the colony size also provides an opportunity to find global optima values. The algorithms are also less sensitive to initialization ranges in obtaining optimal solutions.

Positive values of Selection Intensity in all schemes represent an increase in the population average fitness after selection. Success Rate is an indicative of obtaining a desired function value. All selection schemes favored good individuals by assigning the reproduction rate > 1. Similarly low values of loss of diversity support the avoidance of premature convergence. In general, the ABC algorithms combined with different selection schemes perform better on various parameters. In future work, the performance of the ABC can be improved by hybridizing it with a suitable selection scheme and an effective neighbor search technique.

References

Ajit Kumar, Dharmender Kumar, S.K. Jariai


S.K. Jariai received his Ph.D. from Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). He is Associate Professor in Mechanical Engg. at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). He has to his credit a number of research papers in international journals and conferences. His research interests include quality of service, data mining, software testing, and software engineering.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Dharmender Kumar received his Ph.D. from Guru Jambheshwar University of Science and Technology, Hisar (India). He is Associate Professor in Computer Science and Engg. at Guru Jambheshwar University of Science and Technology, Hisar. He has to his credit a number of research papers in international journals and conferences. His research interests include data mining, data warehousing, swarm intelligence, and quality of service.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.

Ajit Kumar received the M.Tech. (Information Technology) from Guru Gobind Singh Indraprastha University, Delhi (India). He is pursuing the Ph.D. (Computer Science and Engg.) at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (India). His research interests include artificial intelligence, data mining, and data warehousing.