Comparison of different mutation strategies applied to artificial bee colony algorithm

Nadezda STANAREVIC
Faculty of Computer Science
University Megatrend Belgrade
Bulevar umetnosti 29
SERBIA
srna@stanarevic.com

Abstract: - Artificial bee colony (ABC) algorithm is a simple but powerful swarm intelligence optimization algorithm which was successfully applied to a number of problems. In this paper we propose a new approach for extending ABC algorithm based on five mutation strategies “borrowed” from differential evolution (DE) algorithm in order to improve the exploitation process. We compared five different strategies with original ABC algorithm on standard benchmark functions for various numbers of problem variables. The experimental results show that the modified ABC algorithms are effective and outperform the original algorithms in most cases.

Key-Words: - Artificial bee colony, Differential evolution, Large-scale optimization problems, Swarm intelligence

1 Introduction

In recent years many nature inspired algorithms have been introduced. An ant colony, a flock of birds or an immune system are typical examples of a swarm system [1]. These algorithms can be classified into different groups depending on the criteria being considered, such as population based, iterative based, stochastic, deterministic, etc.

Optimization is the process of finding the best way to use available resources, while at the same time not violating any of the conditions that are imposed. Users generally demand that a practical minimization technique should fulfill four requirements [2]:

- Ability to handle non-differentiable, nonlinear and multimodal cost functions.
- Parallelizability to cope with computation intensive cost functions.
- Ease of use, i.e. few control variables to steer the minimization. These variables should also be robust and easy to choose.
- Good convergence properties, i.e. consistent convergence to the global minimum in consecutive independent trials.

Several approaches have been proposed to model the specific intelligent behaviors of honey bee swarms and were applied to solving combinatorial type problems [3], [4], [5], [6], [7]. Since its invention by Karaboga in 2005, ABC algorithm has been successfully applied to many kinds of problems [8], [9], [10], [11], [12]. According to the various applications mentioned above, ABC algorithm confirmed its good performances, but we noticed an insufficiency in exploitation process. Inspired by differential evolution (DE) [2], we modified exploitation process by applying different DE mutation strategies. We name the modified ABC algorithm as Differential Evolution Mutation ABC (DEM-ABC).

The rest of this paper is organized as follows. In Sections 2 and 3, ABC and DE are briefly introduced. The modified ABC algorithm called DEM-ABC algorithm is presented in Section 4. Section 5 presents and discusses the experimental results. Finally, the conclusion is drawn in Section 6.

2 ABC algorithm

Artificial bee colony (ABC) is a relatively new member of swarm intelligence. ABC tries to model natural behavior of real honey bees in food foraging. In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees,
onlookers and scouts. The number of employed bees is equal to the number of food sources and an employed bee is assigned to one of the sources.

Short pseudo-code of the ABC algorithm is given below [3]:

1. Initialize the population of solutions
2. Evaluate the population
3. Produce new solutions for the employed bees
4. Apply the greedy selection process
5. Calculate the probability values
6. Produce the new solutions for the onlookers
7. Apply the greedy selection process
8. Determine the abandoned solution for the scout, and replace it with a new randomly produced solution
9. Memorize the best solution achieved so far

An onlooker bee chooses a food source depending on the probability value associated with that food source, \( p_r \), calculated by the following expression:

\[
p_r = \frac{\text{fit}_i}{\sum_{i=1}^{SN} \text{fit}_i}
\]

where \( \text{fit}_i \) is the fitness value of the solution \( i \) which is proportional to the nectar amount of the food source in the position \( i \).

In order to produce a candidate food position from the old one in memory, the ABC uses the following expression:

\[
\nu_{t,j} = x_{i,j} + \phi_{t,j} (x_{j,best} - x_{i,j})
\]

where \( k \in \{1, 2, \ldots, SN\} \) and \( j \in \{1, 2, \ldots, D\} \) are randomly chosen indexes. A greedy selection mechanism is employed as the selection operation between the old one and the candidate [3]. Providing that a position cannot be improved further through a predetermined number of cycles, the food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called “limit” for abandonment. In the ABC, the parameter limit is calculated using the formula \( SN*D \), where \( SN \) is the number of solutions and \( D \) is the number of variables of the problem.

### 3 DE algorithm

A particular EA that has been used for multiobjective optimization is differential evolution (DE). DE is a simple yet powerful evolutionary algorithm by Price and Storn that has been successfully used for solving single-objective optimization problems. Differential evolution is a parallel direct search method which utilizes \( NP \) \( D \)-dimensional parameter vectors \( x_{i,G} \), \( i = 1, 2, \ldots, NP \) as a population for each generation \( G \). \( NP \) does not change during the minimization process. Like other evolutionary algorithms, it starts with an initial population vector, which is randomly generated when no preliminary knowledge about the solution space is available [13]. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector — mutation [2]. A candidate replaces the parent only if it is better than its parent. DE guides the population towards the vicinity of the global optimum through repeated cycles of mutation, crossover and selection. General procedure of the DE algorithm is shown in Fig.1.

![Fig.1 General Evolutionary Algorithm Procedure](image)

During the mutation process for each individual \( X \) in generation \( t \) an associated mutant individual \( Y \) can be created by using one of the mutation strategies [14], [15]. The basic DE algorithm initializes individual solution with random positions in the search-space. For each solution \( x \) three solutions \( x_{t(1)}, x_{t(2)}, x_{t(3)} \) have to be picked from the population at random. Selected solutions must be distinct from each other as well as from current solution \( x \). A general notation for DE algorithm is \( \text{DE}x/y/z \) where \( x \) specifies the base vector to be mutated, \( y \) is the number of difference vectors used, and \( z \) denotes the crossover scheme. The basic mutation strategy is shown as \( \text{rand}/1 \) :

\[
\text{rand}/1: y_{i,j}^{t} = x_{i(1)}^{t} + F*(x_{t(2)}^{t} - x_{t(3)}^{t})
\]

where \( r[k] k \in \{1, 2, \ldots, 5\} \) is a uniformly distributed random integer number in the range \( \{1, NP\} \), \( j \in \{1, 2, \ldots, n\} \), \( F \in [0,2] \) is an amplification factor that controls the rate at which the population evolves. \( F \) has been experimentally determined for each mutation strategy as shown in Table 2.

In the next equation \( \text{best}/1 \) mutant solution \( y \) is created by using two randomly selected solutions \( x_{t(1)}, x_{t(2)}, \) and the best individual \( x_{best,j} \) in the population at generation \( t \):

\[
\text{best}/1: y_{i,j}^{t} = x_{best,j}^{t} + F*(x_{t(1)}^{t} - x_{t(2)}^{t})
\]
Highly beneficial methods \textit{currenttobest}/1, \textit{best}/2, and \textit{rand}/2 use two additional solutions compared to the basic mutation strategy \textit{rand}/1. The usage of two additional different solutions seems to improve the diversity of the population which affects the search directions.

\textit{currenttobest}/1:
\[
y'_{i,j} = x'_{i,j} + F(x'_{\text{best},j} - x'_{i,j}) + F(x'_{(1),j} - x'_{i,j})
\]
\textit{best}/2:
\[
y'_{i,j} = x'_{i,j} + F(x'_{(1),j} - x'_{i,j}) + F(x'_{(2),j} - x'_{i,j})
\]
\textit{rand}/2:
\[
y'_{i,j} = x'_{(1),j} + F(x'_{(2),j} - x'_{i,j}) + F(x'_{(4),j} - x'_{i,j})
\]

Compared to \textit{rand}/1 and \textit{rand}/2, mutation strategies such as \textit{best}/1, \textit{currenttobest}/1 and \textit{best}/2 benefit from their fast convergence by incorporating best solution information in the mutation strategies. The main problem with the use of the best solution information is premature convergence due to the resulting decreased population diversity. In view of the fast but less aggressive convergence performance \textit{currenttobest}/1 mutation strategy uses parent solution \( x_{i,j} \) (current solution) in the process of creating the associated mutant solution \( y \).

4 DEM-ABC algorithm

In optimization algorithms, the exploration refers to the ability to investigate the unknown regions while the exploitation refers to the ability to apply the knowledge of the previous good solutions to find better solutions [16]. According to the Eq. (2) the new candidate solution is generated by moving the old solution towards (or away from) another solution selected randomly from the population. The randomly selected solution can be a good or a bad one, so the new candidate solution is not necessarily a better solution than the previous one. In this paper, we propose a new way of extending ABC to be suitable for solving large-scale unconstrained optimization problems. In order to improve the exploitation we have replaced expression Eq. (2) with one of the mutation strategies mentioned before.

We compared different strategies for creating new solution in onlooker and employed bee phase. Although DEM-ABC uses mutation strategies “borrowed” from DE algorithm, it is still much closer to the ABC algorithm. DEM-ABC uses same random process to initialize population, same expression for calculating probabilities Eq. (1), and same scout mechanism.

The use of the best solution in onlooker and employed bee phase can drive the new candidate solution towards the global best solution; therefore, the exploitation of ABC algorithm can be increased.

Note that the parameter \( F \) in Fig. 2 plays an important role in balancing the exploration and exploitation of the candidate solution search.

5 Experimental study

Five unconstrained benchmark test functions are used to validate the proposed DEM-ABC algorithm: Sphere, Rosenbrock, Griewank, Rastrigin and Schwefel.

\textit{Sphere function} is continuous, convex and unimodal. Global minimum value for this function is 0 and optimum solution is \( x=(0, 0, \ldots, 0) \). Surface plot is shown in Fig. 2.

Definition:
\[
f(x) = \sum_{i=1}^{n} x_i^2
\]

where \( x \) is in the interval of [-100, 100]

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{sphere_function.png}
\caption{Sphere function}
\end{figure}

\textit{Rosenbrock function} has the global optimum inside a long, narrow, parabolic shaped flat valley. Global minimum value for this function is 0 and optimum solution is \( x=(1,1,\ldots,1) \).

Definition:
\[
f(x) = \sum_{i=1}^{n-1} \left[ 100(x_i^2 - x_{i+1}^2)^2 + (x_i - 1)^2 \right]
\]

where \( x \) is in the interval of [-50, 50]. Surface plot is shown in Fig. 3.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{rosenbrock_function.png}
\caption{Rosenbrock function}
\end{figure}
Griewank’s value is 0, and its global minimum is 
\((0,0,\ldots,0)\). Since the number of local optima 
increases with the dimensionality, this function is 
strongly multimodal. The multimodality disappears 
for sufficiently high dimensionalities \((n > 30)\) and 
the problem seems unimodal.

Definition:
\[
f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(x_i / \sqrt{i}) + 1 \quad (6)
\]
where \(x\) is in the interval of \([-600, 600]\). Surface 
plot is shown in Fig. 4.

\[\text{Fig. 4 Griewank function}\]

Rastrigin function is based on Sphere function 
with the addition of cosine modulation to produce 
many local minima. Thus the function is 
multimodal. The global minimum value for this 
function is 0 and the corresponding global optimum 
solution is \(x = (0,0,\ldots,0)\).

Definition:
\[
f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i)) \quad (7)
\]
where \(x\) is in the interval of \([-5.12, 5.12]\). Surface 
plot is shown in Fig. 5.

\[\text{Fig. 5 Rastrigin function}\]

Schwefel function has several local minima. The 
global minimum value for this function is \(-418.9829\) 
and the corresponding global optimum 
solution is \(x = (1,1,\ldots,1)\). The surface of Schwefel 
function is composed of a great number of peaks 
and valleys. The function has a second best 
minimum far from the global minimum where many 
search algorithms are trapped.

Definition:
\[
f(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \quad (8)
\]
where \(x\) is in the interval of \([-500, 500]\). Surface 
plot is shown in Fig. 6.

\[\text{Fig. 6 Schwefel function}\]

Testing results will be presented in the following 
section.

5.1. Tests and results
For all benchmark functions we set the parameters 
as shown in Table 1:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max eval. fun. calls</td>
<td>1000000</td>
</tr>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Limit</td>
<td>(Sn<em>D</em>0.5)</td>
</tr>
</tbody>
</table>

Table 1 – Parameters values for DEM-ABC

The values of amplification factor (F) are shown in 
Table 2. These values are based on empirical 
experiments.

<table>
<thead>
<tr>
<th>Mutation strategy</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 2 – Values of amplification factor (F) for 
different mutation strategies

The proposed DEM-ABC algorithm is coded in 
C++ and run on a Pentium Core2Duo, 3-GHz 
computer with 6 GB RAM memory and Microsoft 
Visual Studio 2010.

Control parameters are:
1. Bee Num \(NP\) is number of bees in the colony 
   (employed bees plus onlooker bees).
2. Limit controls the number of trials to improve 
certain food source. If a food source could not
be improved within defined number of trial, it is abandoned by its employed bee.

3. Max evaluation function calls: maximum number of objective function calls

Comparison has been made between DE [17], PSO [18], original ABC algorithm [19] and our modified ABC algorithm for dimensions $D = 10$, 100 and 500. The results for modified ABC algorithm (DEM-ABC) are given for all five mutation strategies. Common parameters such as population number, maximum evaluation number were set at same values for all algorithms.

Population size was set to 100, and Max evaluation function calls was set to 100000 for all functions. Each of the experiments was conducted 30 times using different random seeds.

Tables 3-5 show the optimization results of the Sphere, Schaffer, Rosenbrock, Rastrigin and Griewank large scale function respectively. It can be observed that the performances of DEM-ABC algorithm with mutation strategy 4 are superior to ABC algorithm for all values of the parameter $D$.

As shown in Table 5, all DEM-ABC algorithms achieved better results than ABC algorithm for all the functions except for the Rosenbrock function, where mutation 1, 2 and 3 reached slightly worse results.

6 Conclusion

In this paper, we compared the performance of our proposed DEM-ABC algorithm with the original ABC, DE and PSO on a set of 5 large-scaled unconstrained benchmark functions. These algorithms are chosen because they are also swarm intelligence and population based algorithms as the ABC algorithm. Our suggested modification improves the performance of ABC algorithm in terms of improving the exploitation process in employed and onlooker phase. The experimental results of different types of mutation strategies showed that the DEM-ABC algorithm is effective and powerful algorithm for unconstrained large-scaled optimization problems.

<table>
<thead>
<tr>
<th>$D = 10$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>PSO</td>
<td>DE</td>
<td>ABC</td>
<td>DEM-ABC1</td>
<td>DEM-ABC2</td>
<td>DEM-ABC3</td>
<td>DEM-ABC4</td>
</tr>
<tr>
<td>Sphere</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Schwefel</td>
<td>-2654.03</td>
<td>-4177.99</td>
<td>-4189.83</td>
<td>-240.830</td>
<td>-240.830</td>
<td>-240.830</td>
<td>-240.830</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>0.426</td>
<td>0.000</td>
<td>0.013</td>
<td>0.021</td>
<td>0.779</td>
<td>0.073</td>
<td>0.006</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>7.363</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Griewank</td>
<td>0.059</td>
<td>0.004</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 3. Best results of PSO, DE, ABC and DEM-ABC algorithms on unconstrained large-scale benchmark problems for $D=10$

<table>
<thead>
<tr>
<th>$D = 100$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>PSO</td>
<td>DE</td>
<td>ABC</td>
<td>DEM-ABC1</td>
<td>DEM-ABC2</td>
<td>DEM-ABC3</td>
<td>DEM-ABC4</td>
</tr>
<tr>
<td>Sphere</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Schwefel</td>
<td>-20100.40</td>
<td>-31182.50</td>
<td>-41898.30</td>
<td>-2408.301</td>
<td>-2408.301</td>
<td>-2408.301</td>
<td>-2408.301</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>113.144</td>
<td>132.349</td>
<td>0.055</td>
<td>0.002</td>
<td>0.015</td>
<td>91.131</td>
<td>0.011</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>148.249</td>
<td>133.114</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>42.232</td>
<td>0.000</td>
</tr>
<tr>
<td>Griewank</td>
<td>0.049</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 4. Best results of PSO, DE, ABC and DEM-ABC algorithms on unconstrained large-scale benchmark problems for $D=100$

<table>
<thead>
<tr>
<th>$D = 500$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>PSO</td>
<td>DE</td>
<td>ABC</td>
<td>DEM-ABC1</td>
<td>DEM-ABC2</td>
<td>DEM-ABC3</td>
<td>DEM-ABC4</td>
</tr>
<tr>
<td>Sphere</td>
<td>181.160</td>
<td>20.330</td>
<td>8.7E-7</td>
<td>1.2E-07</td>
<td>2.8E-11</td>
<td>2.8E-08</td>
<td>2.7E-07</td>
</tr>
<tr>
<td>Schwefel</td>
<td>-98.1E+03</td>
<td>-13.1E+04</td>
<td>-19.1E+04</td>
<td>-12.1E+03</td>
<td>-12.1E+03</td>
<td>-12.1E+03</td>
<td>-12.1E+03</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>10.9E+05</td>
<td>87.2E+09</td>
<td>10.1E+02</td>
<td>2.2E+03</td>
<td>1.7E+03</td>
<td>1.2E+03</td>
<td>1.9E+02</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>103.040</td>
<td>594.690</td>
<td>87.960</td>
<td>1.9E-03</td>
<td>2.9E-01</td>
<td>2.5E-02</td>
<td>7.8E-01</td>
</tr>
<tr>
<td>Griewank</td>
<td>2.200</td>
<td>0.645</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 5. Best results of PSO, DE, ABC and DEM-ABC algorithms on unconstrained large-scale benchmark problems for $D=500$
References:


