Incremental Enhanced Artificial Bee Colony Algorithm with Local Search

Neha Pathak
Research Scholar, Computer Science Dept.
ITM UNIVERSE
Gwalior, India

Manuj Mishra
Asst. Professor, Computer Science Dept.
ITM UNIVERSE
Gwalior, India

Shiv Pratap Singh Kushwah
Asst. Professor, Computer Science Dept.
ITM UNIVERSE
Gwalior, India

ABSTRACT

Swarm intelligence is the collective problem solving behavior of groups of artificial agents. These agents local interaction with each other can be negative, positive or neutral. Here positive interaction helps agents to solve a problem while negative interaction block the agents for solving problem and neutral interaction does not affect the swarm’s performance. In this work, incremental enhanced ABC algorithm with local search is used for reducing negative interaction without complexifying the agent’s behavior. Here in the enhanced artificial bee colony algorithm, one additional phase in the form of mutation operator is used. With the help of mutation operator, algorithm may not be trapped into local optima due to the chance of changing local best position. The experimental results show that the performance of proposed algorithm and the proposed algorithm is compared with standard ABC algorithm, Artificial Bee Colony algorithm with mutation algorithm.

Keywords
Artificial Bee Colony, ABC, Genetic Algorithm, Mutation, incremental social learning, Swarm Intelligence.

1. INTRODUCTION

The problem of optimization is one of the most crucial problems now a day and lots of work have been done previously to solve these problems. There are many types of optimization algorithm like GA, ABC, PSO, ACO[4], etc. and lots of works in these algorithms are available in the literature. In paper [1], Dervis Karaboga defined an algorithm, called Artificial Bee Colony (ABC) for solving numerical optimization problem. This algorithm is based on the foraging behavior of honey bees. Different versions of ABC algorithm have been proposed like ABC with mutation , ABC with crossover, ABC with SPV , ABC with crossover and mutation and many different versions have been proposed. In paper [2], dynamically division of bees into subgroup and the searching process is performed on these generated subgroups. In paper [3], Karaboga defines the modified algorithm for solving real parameter optimization problems and comparative study on different evolutionary algorithms like particle swarm optimization (PSO), ABC algorithm, Differential evolution (DE), Ant colony algorithm (ACO) etc and performance derived on different benchmark functions, are discussed on paper [5,7]. For improving the exploitation, karaboga proposed a modified ABC algorithm based on global best solution in paper [6]. D. Hajjun and F. Qiaxian proposed a modified artificial bee colony algorithm for function optimization in paper [8].

In paper [10], travelling salesman problem using bee colony with SPV rule is used. In this paper, incremental approach is used in enhanced ABC algorithm for improving local search.

The overall paper is organized as follows. Section 2 describes the artificial bee colony algorithm with mutation. In section 3, incremental Artificial Bee Colony algorithm with local search is explained. Travelling salesman problem is discussed in section 4. Section 5 describes experimental results and parameter setup for result comparison. Finally, section 6 concludes the paper.

2. ARTIFICIAL BEE COLONY ALGORITHM WITH MUTATION

In paper [9], one more phase in the form of mutation operator is added to original Artificial Bee Colony algorithm. In standard ABC algorithm, there are only 4 phases that described the overall working of this algorithm, but here one additional phase after the scout bee phase of ABC algorithm is added in the form of Mutation operator. Now modified artificial bee colony algorithm has five phases: first initialization phase then employed bee phase, onlooker bee phase, scout bee phase and finally mutation phase. For the local search, employed bee phase is used and the mutation phase is used to find out the new search area of the solution space. With the help of mutation operator, there may be a possibility to change the local best position and the algorithm may not be trapped into local optima. In this work, the mutation phase is implemented on the probabilistic way in each iteration for searching food source during the life process of ABC optimization technique. Food sources are selected randomly from the food size to perform mutation operation. In mutation phase, if generated offspring’s fitness is greater than the older one then replaces the older offspring’s from the new one. Uniform mutation is used in this work.

The overall algorithm is described in following steps:

- Initialization phase.
- REPEAT
  (a) In the Memory, Employed bees are placed on the food sources;
  (b) Generate new offspring from older offspring after Applying mutation operator.
  (c) In the memory, onlooker bees are placed on the food sources;
  (d) For finding new food sources, Send the scout bee to the search area
- UNTIL (requirements not met).

3. INCREMENTAL ABC ALGORITHM WITH LOCAL SEARCH

In the original artificial bee colony algorithm [11], food source position represents a solution and its nectar amount represents fitness value. The employed and onlooker bee
phase finds better food position and scout bee phase is used to generate new food sources if all the food sources exhausts.

In the incremental artificial bee colony algorithm, Food sources are increased with a control parameter ‘g’, that means population size is increased in incremental ABC algorithm. In this, for every ‘g’ iterations, a new food source is increased until maximum number of food sources is reached. IABC algorithm begins with some food sources and placed new food sources towards best-so-far solution. The above concept is implemented as –

\[ X_{\text{new},i} = X_{\text{new},i} + \text{rand}(0,1) \times \left( X_{\text{gbest},i} - X_{\text{new},i} \right) \] (1)

Where \( X_{\text{new},i} \) = new food source locations generated randomly

\( X_{\text{gbest},i} \) = best-so-far locations of food sources

Similarly in scout bee phase. Replacement policies are used like above by using a control parameter ‘R_factor’. This parameter controls the closeness of new food source with best-so-far food source. The above concept is implemented as-

\[ X_{\text{new},i} = X_{\text{gbest},i} + \text{R_factor} \times \left( X_{\text{gbest},i} - X_{\text{new},i} \right) \] (2)

The above modifications improve the exploit behavior and help to converge towards optimal solution.

Basic steps of incremental ABC algorithm are given below-

1. Initialize number of food sources
2. REPEAT
   If Failure_attempts = Failure_Max then
     Perform local search on the food sources selected randomly.
   Else
     Perform local search on the best food source
3. Employed bee phase
4. Scout bee phase
   If new food source addition criteria met then
     Add new food source
5. UNTIL (termination criteria not met)

4. TRAVELLING SALESMAN PROBLEM

The travelling salesman (TSP) problem is described as follows: Given a list of cities and a travelling cost between each pair of cities. Here the TSP problem is that to find the shortest route with minimum cost between all the cities and returning to starting position.

In figure 1, there are 7 cities which are connected and A is the source node or starting position. Here from the starting node, visit all the cities with minimum cost. Cost between two cities is not mentioned here. It may vary from problem to problem.

To calculate the minimum cost between all the cities from initial node, following formula is used.

\[ \text{Cost} = \sum_{i=1}^{n} C_{ij} \]

5. PROPOSED WORK

In this proposed work, incremental modified artificial bee colony algorithm with local search is used. In the enhanced artificial bee colony algorithm, one additional phase in the form of mutation operator is used. Here mutation operator is used after the employed bee phase of original ABC algorithm. Now modified ABC algorithm has five phases: Initialization phase, employed bee phase, mutation phase, onlooker bee phase and scout bee phase. When performing mutation, select randomly one of the food sources and replace its one of the dimension value by random number generated in between lower and upper bound value of the food source. The incremental approach that is used in this work for improving local search is explained in section 3. The proposed algorithm is described below-

1. Initialize number of food sources
2. REPEAT
   If Failure_attempts = Failure_Max then
     Perform local search on the food sources selected randomly.
   Else
     Perform local search on the best food source
3. Employed bee phase
4. Mutation phase
   If new food source addition criteria met then
     Add new food source
5. UNTIL (termination criteria not met)

\[ \text{MUTATION PHASE} \]

If mutation criteria is met then

(a) From current population, select random particle for mutation operator.
6. EXPERIMENTAL RESULTS AND PARAMETER SETUP

For interpreting Travelling salesman problem by using incremental enhanced ABC algorithm with local search, the following control parameters are used –

Maximum cycle number, i.e. MCN = 500, 1000, 1500 and 2000

Colony size = 20.

Each individual Dimension = 10, 20 and 30

Number of Iteration = 30.

Experimental results are shown in fig 1, fig 2 and fig 3 for the dimensions 10, 20 and 30 respectively.

7. CONCLUSION

Artificial bee colony algorithm is one of the optimization algorithms for interpreting different optimization problems which are used to find out the global optima. In this work, enhanced algorithm is used which uses one additional step in the form of mutation operator is used. With this mutation operator, Algorithm may not be trapped into local optima. Finally enhanced ABC algorithm with local search using incremental approach is used and the proposed work is compared with previously defined algorithms and implemented on travelling salesman problem. Future work is to used the proposed work on different types of optimization problem and compare them with older one.

8. REFERENCES


<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MCN = 500</th>
<th>MCN = 1000</th>
<th>MCN = 1500</th>
<th>MCN = 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>88.73</td>
<td>83.21</td>
<td>81.19</td>
<td>79.88</td>
</tr>
<tr>
<td>ABC with mutation</td>
<td>86.90</td>
<td>82.65</td>
<td>80.09</td>
<td>78.15</td>
</tr>
<tr>
<td>Proposed Algorithm</td>
<td>86.12</td>
<td>81.57</td>
<td>79.24</td>
<td>77.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MCN = 500</th>
<th>MCN = 1000</th>
<th>MCN = 1500</th>
<th>MCN = 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>175.56</td>
<td>172.25</td>
<td>169.36</td>
<td>167.65</td>
</tr>
</tbody>
</table>
### Table 3: Comparative Results For Dimension = 30

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MCN = 500</th>
<th>MCN = 1000</th>
<th>MCN = 1500</th>
<th>MCN = 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC with Mutation</td>
<td>174.25</td>
<td>171.11</td>
<td>168.68</td>
<td>166.21</td>
</tr>
<tr>
<td>Proposed Algorithm</td>
<td>173.58</td>
<td>170.65</td>
<td>167.49</td>
<td>165.39</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MCN = 500</th>
<th>MCN = 1000</th>
<th>MCN = 1500</th>
<th>MCN = 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>268.31</td>
<td>265.90</td>
<td>263.29</td>
<td>261.94</td>
</tr>
<tr>
<td>ABC with Mutation</td>
<td>267.93</td>
<td>264.55</td>
<td>262.37</td>
<td>260.82</td>
</tr>
<tr>
<td>Proposed Algorithm</td>
<td>266.56</td>
<td>263.14</td>
<td>261.74</td>
<td>259.47</td>
</tr>
</tbody>
</table>