Path Optimization with Artificial Bee Colony Algorithm in WSN

Arti¹, Deepika²
¹M.Tech Research Scholar, Universal Institute of Technology, GJU, Hisar
²Asst. Prof., Universal Institute of Technology, GJU, Hisar

Abstract
Wireless Sensor Networks consisting of nodes with limited power are deployed to gather useful information from the field. In WSNs it is critical to collect the information in an efficient manner. WSN is applied in routing and difficult power supply area or area that cannot be reached and some temporary situations, which do not need fixed network supporting and it can fast deploy with strong anti-damage. In order to avoid the problem we proposed a new technique called Bio-Inspired mechanism for path optimization. ABC is one of the Bio-inspired mechanisms. This paper defines implementation of WSN based on ABC algorithm is done in terms of packet delivery ratio, end to end delay and throughput. Proposed algorithm can avoid network congestion and then it can prolong the life cycle of the whole network. It optimizes the routing paths, providing an effective multi-path data transmission to obtain reliable communications in the case of node faults. The main goal is to maintain the maximum lifetime of network, during data transmission in an efficient manner. In this work, ABC is also used for optimizing a large set of numerical test functions.

Keywords: WSN, ABC, Path Optimization, Benchmark functions.

1. Introduction
The Wireless Sensor Networks (WSN) is intended for monitoring an environment. The main task of a wireless sensor node is to sense and collect data from a certain domain, process them and transmit it to the sink where the application lies[1].In the wireless sensor networks, queue mechanism is adopted which is used for the transmission of the data packets. If heavy traffic load condition occurs, then queue handling capacity of sensor node is not efficient which may causes a data queue overflow in the sensor nodes, lost. So that efficiency and reliability in transmission of data cannot be obtained. The sensor nodes creates the energy hole in the routing path, this is then created by multiple hops operation which is based on relaying the data packets [2]. Because of this energy holes, the life time of the wireless sensor networks is greatly reduced. Wireless sensor networks can be used for many mission critical applications such as target tracking in battlefields and emergency response. In these applications, reliable and timely delivery of sensory data plays a crucial role for the success of the mission.[1] The major problem with wireless sensor networks is their limited source of energy, the coverage constraint and high traffic load. Routing of sensor data has been one of the challenging areas in wireless sensor network research.[2]

2. Artificial Bee Colony Algorithm
Artificial bee colony algorithm that inspired by foraging behavior of bee by Dervis Karaboga in 2005 [3]. In ABC algorithm, the artificial bee colony contains three groups of bees: employed bees, onlooker bees, scout bees. The search carried out by the artificial bees can be summarized as follows: Employed bees determine food source within the neighborhood of the food source in their memory and then it can prolong the life cycle of the whole network. It optimizes the routing paths, providing an effective multi-path data transmission to obtain reliable communications in the case of node faults. The main goal is to maintain the maximum lifetime of network, during data transmission in an efficient manner. In this work, ABC is also used for optimizing a large set of numerical test functions.

At the beginning of ABC algorithm that generated population of food sources randomly. A food source represents a possible solution. The employed bees produce a modification on the position of the food source in their memory and the nectar amount of a food source corresponds to the quality that represents fitness value of the solution then used it to calculate probability values. While the onlooker bees selected the largest probability values of food source, then onlooker bees produce a modification on the position of the food source. The new food sources were calculated fitness value and were compared with the fitness value of the old food source than the largest fitness value of the food source was recorded as the best solution currently .The best food source was recorded in each iteration until reached the maximum number of iterations.[4][5]
3. Related work

In year 2007[6], Dervis Karaboga performs a work,” A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm In this work, ABC algorithm is used for optimizing multivariable functions and the results produced by ABC, Genetic Algorithm (GA), Particle Swarm Algorithm (PSO) and Particle Swarm Inspired Evolutionary Algorithm (PS-EA) have been compared. The results showed that ABC outperforms the other algorithms.

In year 2009[7], Dervis Karaboga performed a work,” A comparative study of Artificial Bee Colony algorithm”. In this paper ABC is used for optimizing a large set of numerical test functions and the results produced by ABC algorithm are compared with the result of different algorithms. Results show that the performance of the ABC is better than or similar to those of other population-based algorithms with the advantage of employing fewer control parameters.

In year 2011[8], Nadezda Stanarevic performed a work,”Modified artificial bee colony algorithm for constrained problems optimization”. In this paper we propose an improved artificial bee colony algorithm for constrained problems. Since the ABC algorithm for constrained problems does not consider the initial population to be feasible we introduced a modification, besides penalty function and Deb’s rule, in a form of “smart bee” (SB) which uses its historical memories for the location and quality of food sources. This modified SB-ABC algorithm was tested on standard benchmark functions for constrained optimization problems and proved to be better.

In year 2012[9], Johannes M. Dieterich performs a work,” Empirical review of standard benchmark functions using evolutionary global optimization”. In this paper we have employed a recent implementation of genetic algorithms to study a range of standard benchmark functions for global optimization. It turns out that some of them are not very useful as challenging test functions. The latter properties seem to be simulated better by two other types of benchmark functions. One type is designed to be deceptive, exemplified here by Lunacek's function. The other type offers additional advantages of markedly increased complexity and of broad tunability in search space characteristics.

4. Proposed Work

The Three Phases of ABC Based Algorithm

Route discovery phase

Route maintenance phase

Route failure handling

The detailed description of various phases of algorithm is as follows:

4.1. Route Discovery Phase

Route discovery phase uses control packet to discover route from source to destination. The control packets are mobile agents which walk through the network to establish routes between nodes. Route discovery uses two bee agents called Employed Bee (EB) and Onlooker Bee (OB). These two bees are similar in structure but differ in the type of work they perform. An EB is an agent, which establishes the path to the source node, and OB establishes path to the destination. An employed bee is broadcast by the sender and relayed by the intermediate nodes till it reaches the destination. A node receiving an EB for the first time creates a record in its routing table. The record includes destination address, next hop and objective value.

4.2. Route Maintenance Phase

Route Maintenance plays a very important role in WSN’s as the network keeps dynamically changing and routes found good during discovery may turn to be bad due to congestion, signal strength, etc. Hence when a node starts sending packets to the destination using the Probabilistic Route Finding algorithm explained below, it is essential to find the goodness of a route regularly and update the pheromone counts for the different routes at the source nodes. To accomplish this, when a destination node receives a packet, it probabilistically sends a Congestion Update message to the source which informs the source of the REM value for that route. This Congestion Update message also serves an ACK to the source.
4.3. Route Failure Handling Phase

This phase is responsible for generating alternative routes in case the existing route fails. Every packet is associated with acknowledgement; hence if a node does not receive an acknowledgement, it indicates that the link is failed. On detecting a link failure the node sends a route error message to the previous node and deactivates this path by setting the objective value to zero. The previous node then tries to find an alternate path to the destination. If the alternate path exists, the packet is forwarded on to that path else the node informs its neighbors to relay the packet towards source. This continues till the source is reached. On reaching the source, the source initiates a new route discovery phase. Hence artificial bee colony algorithm does not break down on failure of optimal path. This helps in load balancing. That is, if the optimal path is heavily loaded, the data packets can follow the next best paths.

Proposed Algorithm:

1. Create a network having 80 nodes arranged in rectangular manner.
2. Select the source and destination from the network.
3. Transfer the data from source towards destination.
4. Calculate the threshold using ABC and value of network parameter.
5. Select source node as Current node i.e. Current node (Nc)=source node.
6. While (Nd~destination) Repeat
   7. Select a relay node from neighbor say Nd and transfer the data to relay node from current node (Nc).
   8. Initiate counter=1;
   9. While counter <=threshold
      10. If (Nd receives data)
          Then send acknowledgement to Nc. And break;
      Else
          Retransmit data from Nc to Nd and counter=counter+1
      End while
   11. If (counter >threshold)
   12. Then find nearest neighbor as Nc. Go to step 7.
   13. End
   14. Exit

Packet Delivery in between the nodes and detect the selfish node:

After detection of selfish node, alternate path is taken through near neighborhood node for packet delivery:

5. Experiments

Benchmark Functions:

A function is multimodal if it has more than one local optima.
A function is unimodal if it has only one local optimum, and this is global optimum. If a function with n-variable can be written as the sum of the n functions of one variable, then this function is called as separable(S) function. Non-separable function cannot be written in this form because there is interrelation among variables of these functions. Therefore, optimizing non-separable functions is more difficult than optimizing separable ones. The most complex case appears.
when the local optima are randomly distributed in the search space. The dimensionality of the search space is another important factor in the complexity of the problem.

The first function is Sphere function. It is one of the most simple test functions available in the specialized literature. This continuous, convex, unimodal and additively separable test function can be scaled up to any number of variables. It belongs to a family of functions called quadratic functions and only has one optimum in the point \(o=(0,...,0)\). The search range commonly used for the Sphere function is \([-100, 100]\) for each decision variable. It is defined as follow:

\[
\begin{align*}
    f(z) &= \sum_{i=1}^{D} z_i^2 \\
    \text{Where } D &= \text{the dimension and } x = (x_1, x_2, \cdots, x_D) \text{ is a D-dimensional row vector (i.e., a } 1\times D \text{ matrix).}
\end{align*}
\]

The second function is Sum Square Function. This continuous, convex, unimodal and additively separable test function can be scaled up to any number of variables. It belongs to a family of functions called quadratic functions and only has one optimum in the point \(o=(0,...,0)\). The search range commonly used for the Sum Square function is \([-10,10]\) for each decision variable. It is defined as follow:

\[
\begin{align*}
    f(z) &= \sum_{i=1}^{D} z_i^2 \\
    \text{The third function is Trid 6 Function. This continuous, convex, unimodal and non- separable test function can be scaled up to any number of variables. It belongs to a family of functions called quadratic functions and only has one optimum in the point \(o=(0,...,0)\). The search range commonly used for the Trid 6 function is \([-36, 36]\) for each decision variable. It is defined as follow:}
\end{align*}
\]

\[
\begin{align*}
    f(z) &= \sum_{i=1}^{D} (z_i - 1)^2 - \sum_{i=2}^{D} z_i z_i - 1 \\
    \text{The fourth function is Trid 10 Function. This continuous, convex, unimodal and non- separable test function can be scaled up to any number of variables. It belongs to a family of functions called quadratic functions and only has one optimum in the point \(o=(0,...,0)\). The search range commonly used for the Trid 6 function is \([-10, 100]\) for each decision variable. It is defined as follow:}
\end{align*}
\]

\[
\begin{align*}
    f(z) &= \sum_{i=1}^{10} (z_i - 1)^2 - \sum_{i=2}^{10} z_i z_i - 1 \\
    \text{The fifth function is Rastrigin function whose value is 0 at its global minimum \((0,0,...,0)\). Range for the function is \([-5.12,5.12]\). This function is based on Sphere function with the addition of cosine modulation to produce many local minima. Thus, the function is multimodal. The locations of the minima are regularly distributed. The difficult part about finding optimal solutions to this function is that an optimization algorithm easily can be trapped in a local optimum on its way towards the global optimum. It is defined as follow:}
\end{align*}
\]

\[
\begin{align*}
    f(x) &= \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i) + 10) \\
    \text{The sixth function is Griewank function whose value is 0 at its global minimum \((0,0,...,0)\). It is multimodal and non- separable, with several local optima within the search region defined by \([-600, 600]\). The aim is to overcome the failure of the techniques that optimize each variable independently. The optima of Griewank function are regularly distributed. Since the number of local optima increases with the dimensionality, this function is strongly multimodal. It is defined by:}
\end{align*}
\]

\[
\begin{align*}
    f(x) &= \frac{1}{4000} \sum_{i=1}^{D} z_i^2 - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \\
    \text{Settings for ABC algorithm:}
\end{align*}
\]

ABC algorithm has a few control parameters: Maximum number of cycles (MCN) equals to the maximum number of generation and the colony size equals to the population size, i.e. 20. The percentage of onlooker bees was 50% of the colony, the employed bees were 50% of the colony and the number of scout bees was selected as one. The increase in the number of scouts encourages the exploration as the increase of onlookers on a food source increases the exploitation. In the experiments, maximum number of generations were 1,000 for the dimensions 10, 20 and 30, respectively; and the population size was 20. Each of the experiments was repeated 5 times with different random seeds. The mean function values of the best solutions found by the algorithms for different dimensions have been recorded. The mean of the function values obtained by the ABC are given in Table 2.

D, C, Range and Min. in Table 1, are dimensions, characteristics, lower and upper bounds of search spaces and global minimum values of the functions, respectively.
Experiment 1: Performance Analysis on Basic Benchmark Functions.

For each function, all the methods were run 10 times with random seeds on the IBM-compatible PC with Intel Core2 Duo 3.00 Ghz CPU and 4 Gb RAM.

<table>
<thead>
<tr>
<th>Function</th>
<th>D</th>
<th>C</th>
<th>Range</th>
<th>Min</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere(F1)</td>
<td>5</td>
<td>U.S</td>
<td>[-100.100]</td>
<td>0</td>
<td>( f(x) = \sum_{i=1}^{D} x_i^2 )</td>
</tr>
<tr>
<td>Sum Square(F2)</td>
<td>6</td>
<td>U.S</td>
<td>[-10,10]</td>
<td>0</td>
<td>( f(x) = \sum_{i=1}^{D} t x_i^2 )</td>
</tr>
<tr>
<td>Trid 6 (F3)</td>
<td>4</td>
<td>U.N</td>
<td>[-36,36]</td>
<td>-50</td>
<td>( f(x) = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=1}^{D} x_i x_i - 1 )</td>
</tr>
<tr>
<td>Trid 10 (F4)</td>
<td>10</td>
<td>U.N</td>
<td>[-100,100]</td>
<td>-210</td>
<td>( f(x) = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=1}^{D} x_i x_i - 1 )</td>
</tr>
<tr>
<td>Rastrigin (F5)</td>
<td>60</td>
<td>M.S</td>
<td>[-5.12,5.12]</td>
<td>0</td>
<td>( f(x) = \sum_{i=1}^{D} (x_i^2 - 10 \cos(2 \pi x_i) + 10) )</td>
</tr>
<tr>
<td>Griewank (F6)</td>
<td>50</td>
<td>M.N</td>
<td>[-600,600]</td>
<td>0</td>
<td>( f(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 )</td>
</tr>
</tbody>
</table>

Table 2 The best, worst and mean results obtained by the ABC on the benchmark functions after 10 runs.

<table>
<thead>
<tr>
<th>Function</th>
<th>D</th>
<th>ABC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Best</td>
</tr>
<tr>
<td>F_1</td>
<td>5</td>
<td>1.77461</td>
</tr>
<tr>
<td>F_2</td>
<td>6</td>
<td>7.60560</td>
</tr>
<tr>
<td>F_3</td>
<td>4</td>
<td>0.64392</td>
</tr>
<tr>
<td>F_4</td>
<td>10</td>
<td>0.41008</td>
</tr>
<tr>
<td>F_5</td>
<td>60</td>
<td>1.44562</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1.23792</td>
</tr>
<tr>
<td>F_6</td>
<td>50</td>
<td>0.00045</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0.00047</td>
</tr>
</tbody>
</table>
Experiment 2- Performance Analysis Based on CEC2005 Benchmark Functions

The benchmark functions (F1–F6) in this experiment have been proposed for testing the performance of the algorithms presented in CEC2005 special session on real parameter optimization. This test set contains 6 functions composed of different characteristics such as unimodal, multimodal, Seperable, Non- Seperable [10].

Table 3. The Result of ABC for F1 to F6 functions, dimension=10, Colony size=50, cycle=1000.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>3.4841</td>
<td>10.2704</td>
<td>0.4550</td>
<td>0.2884</td>
<td>1.4456</td>
<td>0.0007</td>
</tr>
<tr>
<td>Worst</td>
<td>4.8396</td>
<td>24.3501</td>
<td>0.9264</td>
<td>1.8667</td>
<td>4.3224</td>
<td>0.0014</td>
</tr>
<tr>
<td>Mean</td>
<td>4.1847</td>
<td>14.8565</td>
<td>0.7609</td>
<td>1.1399</td>
<td>3.5549</td>
<td>0.0010</td>
</tr>
</tbody>
</table>

Table 4. The Result of ABC for F1 to F6 functions, dimension=10, Colony size=100, cycle=1000.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>1.7778</td>
<td>8.0048</td>
<td>0.4986</td>
<td>0.4367</td>
<td>2.0240</td>
<td>0.0008</td>
</tr>
<tr>
<td>Worst</td>
<td>6.0920</td>
<td>12.7715</td>
<td>1.5704</td>
<td>1.1573</td>
<td>3.7953</td>
<td>0.0011</td>
</tr>
<tr>
<td>Mean</td>
<td>3.4891</td>
<td>10.1606</td>
<td>1.1433</td>
<td>0.9320</td>
<td>2.8363</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

Table 5. The Result of ABC for F1 to F6 functions, dimension=10, Colony size=200, cycle=1000.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>2.5879</td>
<td>16.8929</td>
<td>0.8048</td>
<td>0.6589</td>
<td>2.3464</td>
<td>0.0006</td>
</tr>
<tr>
<td>Worst</td>
<td>4.4329</td>
<td>67.5959</td>
<td>1.3859</td>
<td>1.3081</td>
<td>5.1443</td>
<td>0.0012</td>
</tr>
<tr>
<td>Mean</td>
<td>3.6845</td>
<td>29.2365</td>
<td>1.0714</td>
<td>0.9471</td>
<td>3.5244</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

Table 6. The Result of ABC for F1 to F6 functions, dimension=10, Limit=100, Max. Eval: 1000, MM: Multimodal, UM: Unimodal

<table>
<thead>
<tr>
<th></th>
<th>F1 (UM)</th>
<th>F2 (UM)</th>
<th>F3 (UM)</th>
<th>F4 (UM)</th>
<th>F5 (MM)</th>
<th>F6 (MM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>1.4294</td>
<td>4.3691</td>
<td>0.4409</td>
<td>0.5573</td>
<td>2.4710</td>
<td>0.0006</td>
</tr>
<tr>
<td>Worst</td>
<td>4.4918</td>
<td>18.0277</td>
<td>1.3019</td>
<td>1.3939</td>
<td>6.1680</td>
<td>0.0012</td>
</tr>
<tr>
<td>Mean</td>
<td>3.2360</td>
<td>10.7408</td>
<td>0.7932</td>
<td>0.9887</td>
<td>3.9685</td>
<td>0.0009</td>
</tr>
</tbody>
</table>
Table 7. The Result of ABC for F1 to F6 functions, dimension=20, Limit =200, Max. Eval.:1000,

<table>
<thead>
<tr>
<th></th>
<th>F1(UM)</th>
<th>F2 (UM)</th>
<th>F3(UM)</th>
<th>F4 (UM)</th>
<th>F5(MM)</th>
<th>F6(MM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>3.4744</td>
<td>8.0416</td>
<td>0.4100</td>
<td>0.6323</td>
<td>2.2288</td>
<td>7.8741</td>
</tr>
<tr>
<td>Worst</td>
<td>5.5610</td>
<td>18.8834</td>
<td>1.5469</td>
<td>1.1485</td>
<td>4.3432</td>
<td>12.2022</td>
</tr>
<tr>
<td>Mean</td>
<td>4.1499</td>
<td>13.1303</td>
<td>0.8728</td>
<td>0.9277</td>
<td>3.3054</td>
<td>9.9705</td>
</tr>
</tbody>
</table>

Table 8. The Result of ABC for F1 to F6 functions, dimension=30, Limit =300, cycle=1000.

<table>
<thead>
<tr>
<th></th>
<th>F1(UM)</th>
<th>F2 (UM)</th>
<th>F3(UM)</th>
<th>F4 (UM)</th>
<th>F5(MM)</th>
<th>F6(MM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>2.1480</td>
<td>10.9257</td>
<td>0.5569</td>
<td>0.8395</td>
<td>2.0056</td>
<td>6.5430</td>
</tr>
<tr>
<td>Mean</td>
<td>3.6497</td>
<td>12.5926</td>
<td>10.5210</td>
<td>1.2119</td>
<td>3.4922</td>
<td>8.4810</td>
</tr>
</tbody>
</table>

6. Result Validation

Here the detailed implementation of proposed approach is presented. Simulation is done using the network simulator tool MATLAB. The Following Performance Evaluation matrices are used to calculate the performance of the network:

i) Throughput:

Throughput of network is defined as total number of packets received at each destination node divided by total number of packets transmitted by each source node over the network.

ii) Loss-rate:

Loss-rate of the network indicates number of packets dropped during transmission. It is calculated as total number of packets dropped per second in the network.

iii) Link delay:

It is the time taken by the link to transfer a packet from the source node to destination node.

iv) Packet delivery ratio:

Packet delivery ratio is the ratio of no of packets received to no. of packets generated.

7. Graphs

In order to show the performance of the ABC algorithm more clearly, the graphical representations of the results in Table 1 are reproduced in Figs. 1–7.

---

**Fig 1: Graph for Best value of Functions**
Fig 2: Graph for Mean value of functions.

Fig 3: Effect of limit on UniModal Functions.

Fig 4: Effect of limit on MultiModal Functions.

Fig 5: Packet delivery ratio.

Fig 6: End to End delay.

Fig 7. Throughput.
Conclusion

In this paper, we presented a new protocol for WSN path optimization Operations. The protocol is achieved by using ABC algorithm to optimize routing paths, providing an effective multi-path data transmission to obtain reliable communications in the case of node faults. We aimed to maintain network life time in maximum, while data transmission is achieved efficiently. Our study was concluded to evaluate the performance of artificial bee colony based algorithm in terms of Packet Delivery Ratio, Average end-to-end delay and throughput. Our proposed algorithm can control the overhead generated by bees, while achieving faster end-to-end delay and improved packet delivery ratio. The capability of the ABC algorithm for constrained optimization problems was investigated through the performance of several experiments on well-known test problems. ABC algorithms were tested on six high dimensional numerical benchmark functions that have multimodality. From the simulation results it was concluded that the proposed algorithm has the ability to get out of a local minimum and can be efficiently used for multivariable, multimodal function optimization. In future, With the experience obtained from the real-time implementation and testing we will improve this algorithm. And will include investigation of the ABC performance in other benchmark and real life problems. The main steps in further modifications of ABC algorithm for constrained problems are directed towards finding better feasible solutions that will guide the swarm towards the optimum solution.

References


