Improved Swarm Bee Algorithm for Global Optimization

Tarun Kumar Sharma
Department of Paper Technology
Indian Institute of Technology
Roorkee, India

Millie Pant
Department of Paper Technology
Indian Institute of Technology
Roorkee, India

ABSTRACT
Artificial Bee Colony (ABC) algorithm simulates the foraging behavior of honey bee colonies. ABC is an optimization technique, which is used in finding the best solution from all feasible solutions. However, there is still an insufficiency in ABC regarding improvement in exploitation and convergence speed. In order to improve the performance of ABC we embedded PSO into ABC. As PSO has memory, knowledge of good solutions is retained by all the particles. In addition, to improve the convergence speed, the initial population of food sources is produced using the union of random generated population using random numbers and chaotic systems. This modification in basic ABC results in new search mechanism, ISBC (Improved Scout Bee Colony). Experiments are conducted on a set of 6 shifted benchmark functions. The results demonstrate good performance of ISBC in solving complex numerical optimization problems when compared with two ABC-based algorithms.

General Terms
Evolutionary Algorithms, Optimization, Performance of ABC, Initial Food Locations in ABC.

Keywords
Artificial Bee Colony, Convergence, Exploration and Exploitation, Chaotic Maps.

1. INTRODUCTION
ABC, inspired by the intelligent foraging behavior of honeybees, was proposed by Karaboga in 2005 [1]. It’s gradually increasing interest of many researchers because of its simplicity, outstanding performance, wide applicability and fewer control parameters. The basic ABC has been compared with other population based evolutionary algorithms, such as GA, PSO, and DE [2][3]. ABC has been used to solve real-world problems such as flow shop scheduling problems [4], minimum spanning tree problems [5], and parameter estimation problems [6]. However, like other evolutionary algorithms ABC also faces some limitations while handling complex multimodal functions, functions having narrow curve valley and its stochastic nature slows down the convergence speed [7]. The reason behind all this is improper balance between exploration and exploitation. ABC is good at exploration but poor at exploitation [8]. NM simplex search has been introduced into ABC, and a hybrid simplex ABC was proposed for inverse analysis to improve the performance of ABC [9].

In this paper we combined ABC & PSO [10-11] global optimization algorithms, and propose the novel hybrid algorithm ISBC. As PSO has memory, knowledge of good solutions is retained by all the particles. Moreover, PSO & ABC both work with an initial population of solutions. So combining the searching abilities of both algorithms seems to be a reasonable approach to improve the convergence speed and to balance exploration & exploitation of basic ABC.

The paper is organized as follows. Section 2 provides the compact reviews of the basic ABC. Chaotic system is presented in Section 3. Section 4 discusses the proposed algorithm. In section 5 experimental settings & performance criterion is defined. The proposed algorithm is tested against a set of 6 shifted benchmark test functions and the results are compared in section 6. Finally, Section 7 summarizes the contribution of this paper along with some future research directions.

2. ARTIFICIAL BEE COLONY
The artificial bee colony classifies the foraging artificial bees into three groups of bees: employed bees, onlookers and scouts. The first half of the colony consists of the employed artificial bees and the second half includes the onlookers. 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 exhausted by the bees becomes a scout. The pseudo-code of the algorithm is given in Fig. 1. Step-by-step procedure of ABC is given as follows:

(a) Randomly distributed food sources are generated over a D-dimensional search space.

(b) An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, P", calculated by the expression:

\[ P_i = \frac{fit_i}{\sum_{j=1}^{NP} fit_j} \]  

Where \( fit_i \) is the fitness value of the solution \( i \) which is proportional to the nectar amount of the food source in the position \( i \) and \( NP \) is the number of food sources which is equal to the number of employed bees. In order to produce a candidate food position from the old one in memory, the ABC uses the following expression:

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

Where \( k \) \( \in \) \{1, 2, ..., NP\} and \( j \) \( \in \) \{1, 2, ..., D\} are randomly chosen indexes. Further, \( k \neq i \) and \( r_{ij} \) is a random number between [-1, 1].

(c) In ABC algorithm, “limit” is an important control parameter, it controls the times of updates of a certain solution. If a solution cannot be improved further through a predetermined number of cycles called limit then that solution is assumed to be abandoned, and the employed bee becomes a...
scout. If the solution is $x_i$ and $j \in \{1, 2, \ldots, D\}$ to be abandoned, then a new solution produced randomly would replace $x_i$ by:

$$x_i = x_i^{\prime} + \text{rand}(0,1)(x_i^{\prime\prime} - x_i^{\prime})$$  \hspace{1cm} (3)

where $X_{\text{min}}^j$ is the lower bound of the parameter $j$ and $X_{\text{max}}^j$ is the upper bound of the parameter $j$.

\begin{verbatim}
Begin
1. Initialize the population of food sources $x_i$, $i = 1, \ldots, SN$
2. Evaluate each food source $x_i$, $i = 1, \ldots, SN$
3. $\text{cycle} = 1$
Repeat
For each food source $x_j$ in the population
4. Generate a new food source $v_j$ by its corresponding employed bee (Eq. 2)
5. Evaluate $v_j$
6. Keep the best solution between $x_i$ and $v_j$
End
7. Select, based on fitness proportional selection, the food sources to be visited by onlooker bees
For each food source $x_j$ chosen by an onlooker bee
8. Generate a new food source $v_j$ by its corresponding onlooker bee (Eq. 2)
9. Evaluate $v_j$
10. Keep the best solution between $x_i$ and $v_j$
End
11. Use the scout bee to replace those abandoned food sources
12. Save in memory the best food source so far
13. $\text{cycle} = \text{cycle} + 1$
Until $\text{cycle}$
End
\end{verbatim}

\begin{figure}

Fig. 1: Pseudocode of ABC

3. CHAOTIC MAPS

Most of the chaotic maps available in the literature possess certainty, ergodicity and the stochastic property. They have also been used together with some heuristic optimization algorithms [12][13] to express optimization variables. The choice of chaotic sequences is justified theoretically by their unpredictability, i.e., by their spread-spectrum characteristic, non-periodic, complex temporal behavior, and ergodic properties. A chaotic map is a discrete-time dynamical system

$$X_{k+1} = f(X_k), \hspace{1cm} 0 < X_k < 1, \hspace{1cm} k = 0,1,2,\ldots$$

running in chaotic state. The chaotic sequence $\{X_k : k = 0,1,2,\ldots\}$ can be used as spread-spectrum sequence like a random number sequence. Chaotic sequences have been proven easy and fast to generate and store, there is no need for storage of long sequences [14]. Merely a few functions (chaotic maps) and few parameters (initial conditions) are needed even for very long sequences. In addition, an enormous number of different sequences can be generated simply by changing its initial condition. Moreover, these sequences are deterministic and reproducible. Recently, chaotic sequences have been adopted instead of random sequences and very interesting and somewhat good results have been shown in many applications such as secure transmission [15][16], and nonlinear circuits [17], DNA computing [18], image processing [19].

The spread spectrum property of chaotic sequences can particularly be useful for population based nature inspired algorithms like that of ABC, which depend a lot on generation of random numbers. The selected chaotic maps that produce chaotic numbers in (0, 1) for the experiments have been listed in Table 1.

4. ISBC: Proposed Algorithm

4.1 Initial Population using Chaotic Map

Population initialization is a crucial task in evolutionary algorithms which affects the convergence speed as well as the quality of the final solution. When no information about the solution is available, then random initialization is the commonly used method to generate candidate initial population. The chaotic maps owes the randomness and sensitivity dependence on the initial conditions of chaotic maps, chaotic maps have been used to initialize the population so that the search space information can be extracted to increase the population diversity in [20]. It is intended to enhance the global convergence. This paper proposes a novel initialization approach which employs union of randomly generated uniform random numbers and chaotic systems to generate initial population. In our case sinusoidal iterator is selected and its equation is defined as follows:

$$ch_m = \sin(\pi x_m), x_m \in (0,1), m = 0,1,2,\ldots, m,$$

where $k$ is the iteration counter and $K$ is the preset maximum number of chaotic iterations. Hence the algorithm given in Fig. 2 is used to generate the initial population.

4.2 Modified Search

Exploration and exploitation are the two important aspects in evolutionary computing paradigms. Exploration is the ability to search the solution space to find promising new solutions, and exploitation is the ability to find the optimum solution in the neighborhood of a good solution. The algorithm is improved to balance between the above aspects. To enrich the searching behavior and to avoid being trapped into local optimum, the memory retaining concept of Particle Swarm is incorporated into the ABC. In PSO, the particles tend to move towards good areas in the search space by the information sharing mechanism. The particle moves to a new position calculated by the velocity updated at each time step $t$.

Further the scout bee phase of the algorithm uses Eq. (4) i.e. randomized localization, instead of using Eq. (3), which increases the convergence speed.

$$X_{s} = x_{s} + \text{rand}(X_{x}, X_{y})$$  \hspace{1cm} (4)

where $x_{1}$, $x_{2}$, $x_{3}$ are randomly three distinct individuals from the population and $x_{s}$ is the best of these individuals.

The Pseudocode of the proposed algorithm is given in Fig. 3.
8. end for
   [Initializing population of Food sources using chaotic
   systems]
9. for i=1 to SN
10. do
11. for j=1 to n
12. do
13. Randomly initialize variables $c_{h_{i,j}} \in (0,1)$, set iteration
   counter N=0
14. for m=1 to N do
   $ch_{m+1,j} = \sin(\pi ch_{m,j})$
15. end for
16. $x_j = x_{\min,j} + ch_m (x_{\max,j} - x_{\min,j})$
17. end for
18. end for
19. end for
20. Set the individual counter i=1, j=1
21. Selecting SN fittest individuals from the set of
   $\{x_i \cup OP_i\}$ as initial population.

Fig 2: Pseudocode to Initialize the Food sources

5. EXPERIMENTAL SETTNGS AND
PERFORMANCE CRITERION

5.1 Experimental Settings
The proposed algorithm is tested on 6 shifted benchmark
problems given taken from literature [21]. ABC and the
proposed variants are implemented on Dev-C++ and the
experiments are conducted on a computer with 2.00 GHz Intel
(R) core (TM) 2 duo CPU and 2- GB of RAM. For each
problem, all the algorithms are independently executed 30
times and the average is recorded. The parameters’ setting is
taken as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100 (i.e SN=50)</td>
</tr>
<tr>
<td>Limit</td>
<td>100</td>
</tr>
<tr>
<td>VTR Value</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Maximum MCN</td>
<td>8000</td>
</tr>
<tr>
<td>$\omega$</td>
<td>0.7</td>
</tr>
<tr>
<td>$C_1$, $C_2$</td>
<td>1.49</td>
</tr>
<tr>
<td>Dimension</td>
<td>10, 30</td>
</tr>
</tbody>
</table>

5.2 Performance Criteria
Mean Fitness, Standard Deviation, and Best: The average
of function fitness value that an algorithm can find, using
predefined MCN, is recorded in each run and then average of
the function fitness values are calculated. Also the average,
standard deviation, best and worst of the fitness values are
also calculated.

MCN: The MCN (Maximum Cycle Number) is recorded
when the VTR is reached before to reach maximum MCN. i.e.
we set the termination criteria as $f_{optimal} - f_{global} \leq \text{VTR}$
and record MCN over 30 runs.

Acceleration rate (AR) in %: This criterion is used to compare
the convergence speeds between ABC, PSO and ISBC. It is
defined as follows:

$$\text{AR} = \frac{\text{MCN}_{\text{random algorithm}} - \text{MCN}_{\text{best algorithm}}}{\text{MCN}_{\text{random algorithm}}} \times 100$$

Success Rate (SR) = \frac{\# \text{of successful runs}}{\text{total runs}} \times 100

End

9. Select, based on fitness proportional selection, the food
   sources to be visited by onlooker bees

For each food source $x_i$ chosen by an onlooker bee
10. Generate a new food source $V_i$ by its corresponding
    employed bee using equation
    $V_i = w V_i + C_1 r_1 (pbest_i - X_i) + C_2 r_2 (gbest_i - X_i)$

where $w$ represents the inertia weight, $C_1$, $C_2$ are learning
factors, $r_1$, $r_2$ are independent random numbers uniformly distributed
in the range [0,1]. $V_i$, $X_i$, $pbest_i$ are the velocity,
position and the personal best of the $i^{th}$ particle in the $j^{th}$
dimension respectively. The gbest, is the $j^{th}$ dimension of best
particle in the swarm.

11. Evaluate $V_i$
12. if $(V_i) \leq (X_i)$ then
13. $X_i = V_i$

End

Until cycle
21. End

Fig 3: Proposed Algorithm: ISBC

Table 1: Definitions of various chaotic maps

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic</td>
<td>$X_{n+1} = 4X_n (1 - X_n)$</td>
</tr>
<tr>
<td>Map</td>
<td></td>
</tr>
<tr>
<td>Circle</td>
<td>$X_{n+1} = X_n + 2 \pi \sin(2 \pi X_n) \mod 1$</td>
</tr>
<tr>
<td>Gauss</td>
<td>$X_{n+1} = \left{ \begin{array}{ll} 0, &amp; X_n \in (0,1), \frac{1}{X_n} \mod 1 \leq 1 \ 1, &amp; \frac{1}{X_n} \mod 1 &gt; 1 \end{array} \right.$</td>
</tr>
<tr>
<td>Map</td>
<td></td>
</tr>
</tbody>
</table>
Henon Map

\[ X_{n+1} = 1 - 1.4X_n^2 + 0.3X_{n-1} \]

Sinus Map

\[ X_{n+1} = 2.3(X_n)^2 \sin(\pi X_n) \]

Sinusoidal Iterator

\[ X_{n+1} = \sin(\pi X_n) \]

Tent Map

\[ X_{n+1} = \begin{cases} X_n / 0.7, & \text{if } X_n < 0.7 \\ 10 / 3X_n(1 - X_n), & \text{Otherwise} \end{cases} \]

6. SIMULATION RESULTS

In Table-4 we have recorded the results on the basis of average error. In this case MCN is fixed at 8000, to estimate the average of minimum fitness function value in 30 runs. From the Table-4 it can be clearly observed that for all the 6 shifted benchmark functions ISBC gives better results than ABC and PSO.

In the Table-2, we fixed VTR as given in experimental setting and then calculated the MCN of 30 runs. From Table-2 we can see that the proposed ISBC gives the better results for every function in the comparison to the other algorithms. Further it from Table-2 it is clear that the proposed ISBC is faster than PSO by 40% and ABC by 26%, when D taken as 10 and in other case when D taken as 30, ISBC again performs faster than PSO by 32% and ABC by 16%. The success rates of all the algorithms are given in Table 3, which again demonstrates the efficiency of the proposed algorithm.

MCN taken to estimate the average of minimum fitness function value in 30 runs are also presented graphically in Fig. 4.

7. CONCLUSION AND FUTURE WORK

In the present work we have proposed an algorithm, ISBC to improve the convergence speed and to balance the exploration and exploitation. In this paper we made changes in both employed bee phase as well as the scout bee phase. In addition to this the initial population is generated by the combination of uniformly generated random numbers (rand()) & using chaotic systems. The proposed algorithm is tested on a small bed of shifted benchmark functions. Numerical results indicate that the proposed enhancements improve the performance of basic ABC in terms of convergence rate and fitness function. The future work includes to implementation of this algorithm to real life engineering design problems.

<table>
<thead>
<tr>
<th>Shifted Functions</th>
<th>F</th>
<th>PSO</th>
<th>ABC</th>
<th>ISBC</th>
<th>PSO</th>
<th>ABC</th>
<th>ISBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>F_1</td>
<td>430</td>
<td>379</td>
<td>260</td>
<td>7807</td>
<td>5477</td>
<td>4329</td>
</tr>
<tr>
<td>Schwefel’s</td>
<td>F_2</td>
<td>432</td>
<td>314</td>
<td>196</td>
<td>19901</td>
<td>17493</td>
<td>14968</td>
</tr>
<tr>
<td>Rosenbrock’s</td>
<td>F_3</td>
<td>9810</td>
<td>7639</td>
<td>6814</td>
<td>17495</td>
<td>16019</td>
<td>13714</td>
</tr>
<tr>
<td>Rastrigin’s</td>
<td>F_4</td>
<td>7915</td>
<td>7241</td>
<td>6583</td>
<td>21084</td>
<td>19930</td>
<td>15932</td>
</tr>
<tr>
<td>Griekwank’s</td>
<td>F_5</td>
<td>852</td>
<td>690</td>
<td>469</td>
<td>2161</td>
<td>1545</td>
<td>1367</td>
</tr>
<tr>
<td>Ackley’s</td>
<td>F_6</td>
<td>859</td>
<td>567</td>
<td>378</td>
<td>1891</td>
<td>1393</td>
<td>1185</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shifted Functions</th>
<th>F</th>
<th>PSO</th>
<th>ABC</th>
<th>ISBC</th>
<th>PSO</th>
<th>ABC</th>
<th>ISBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>F_1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Schwefel’s</td>
<td>F_2</td>
<td>20</td>
<td>30</td>
<td>30</td>
<td>20</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Rosenbrock’s</td>
<td>F_3</td>
<td>20</td>
<td>20</td>
<td>30</td>
<td>20</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Rastrigin’s</td>
<td>F_4</td>
<td>30</td>
<td>20</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Griekwank’s</td>
<td>F_5</td>
<td>40</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Ackley’s</td>
<td>F_6</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Table 4: Comparison of proposed ISBC algorithm with PSO, ABC on shifted functions in terms of error (best and mean) and standard deviation (SD) for 10 and 30 dimensions.

<table>
<thead>
<tr>
<th>F</th>
<th>Error Value</th>
<th>PSO</th>
<th>ABC</th>
<th>ISBC</th>
<th>PSO</th>
<th>ABC</th>
<th>ISBC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dimension = 10</td>
<td>Dimension = 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>7.518E-07</td>
<td>7.393E-07</td>
<td>5.279E-07</td>
<td>8.878E-07</td>
<td>8.536E-07</td>
<td>8.192E-07</td>
</tr>
<tr>
<td></td>
<td>SD ($)</td>
<td>1.785E-07</td>
<td>1.030E-07</td>
<td>1.122E-07</td>
<td>1.057E-07</td>
<td>1.022E-07</td>
<td>1.022E-07</td>
</tr>
<tr>
<td></td>
<td>MCN</td>
<td>4.300E+02</td>
<td>3.790E+02</td>
<td>2.600E+02</td>
<td>7.807E+03</td>
<td>5.477E+03</td>
<td>4.129E+03</td>
</tr>
<tr>
<td>$F_2$</td>
<td>best</td>
<td>9.810E-07</td>
<td>9.129E-07</td>
<td>8.093E-07</td>
<td>5.164E+00</td>
<td>2.308E-03</td>
<td>2.092E-03</td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>4.063E-01</td>
<td>8.391E-07</td>
<td>6.926E-07</td>
<td>1.639E+01</td>
<td>3.389E+00</td>
<td>2.912E+00</td>
</tr>
<tr>
<td></td>
<td>SD(±)</td>
<td>5.824E-01</td>
<td>8.184E-07</td>
<td>7.939E-07</td>
<td>7.070E+00</td>
<td>4.285E+00</td>
<td>2.093E+00</td>
</tr>
<tr>
<td></td>
<td>MCN</td>
<td>4.320E+02</td>
<td>3.140E+02</td>
<td>1.960E+02</td>
<td>8.000E+03</td>
<td>8.000E+03</td>
<td>8.000E+03</td>
</tr>
<tr>
<td>$F_3$</td>
<td>best</td>
<td>2.931E+00</td>
<td>2.193E+00</td>
<td>1.294E+00</td>
<td>1.571E+02</td>
<td>1.427E+02</td>
<td>1.379E+02</td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>5.532E+00</td>
<td>4.954E+00</td>
<td>4.001E+00</td>
<td>1.959E+02</td>
<td>1.860E+02</td>
<td>1.549E+02</td>
</tr>
<tr>
<td></td>
<td>SD(±)</td>
<td>1.519E+00</td>
<td>2.937E+00</td>
<td>3.376E+00</td>
<td>1.738E+02</td>
<td>1.688E+02</td>
<td>1.690E+02</td>
</tr>
<tr>
<td></td>
<td>MCN</td>
<td>8.000E+03</td>
<td>7.639E+03</td>
<td>6.814E+03</td>
<td>8.000E+03</td>
<td>8.000E+03</td>
<td>8.000E+03</td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>3.583E-07</td>
<td>3.183E-07</td>
<td>1.039E-07</td>
<td>2.985E-00</td>
<td>1.990E+00</td>
<td>1.099E+00</td>
</tr>
<tr>
<td></td>
<td>SD ($)</td>
<td>4.064E-01</td>
<td>2.000E-05</td>
<td>2.000E-05</td>
<td>2.000E-05</td>
<td>2.000E-05</td>
<td>2.000E-05</td>
</tr>
<tr>
<td></td>
<td>MCN</td>
<td>7.915E+02</td>
<td>6.900E+02</td>
<td>5.683E+02</td>
<td>7.915E+02</td>
<td>7.915E+02</td>
<td>7.915E+02</td>
</tr>
<tr>
<td></td>
<td>SD ($)</td>
<td>6.894E-07</td>
<td>6.081E-07</td>
<td>3.875E-07</td>
<td>8.946E-07</td>
<td>8.946E-07</td>
<td>8.946E-07</td>
</tr>
<tr>
<td></td>
<td>MCN</td>
<td>8.590E+02</td>
<td>5.670E+02</td>
<td>3.780E+02</td>
<td>8.590E+02</td>
<td>8.590E+02</td>
<td>8.590E+02</td>
</tr>
</tbody>
</table>

Fig 4: MCN taken by 6 Shifted benchmark functions with D=10 and 30 by PSO, ABC and ISBC

8. REFERENCES


