# Dynamic Swarm Artificial Bee Colony Algorithm

**Article** · January 2012

DOI: 10.4018/jaec.2012100102



**4 authors:**

Harish Sharma
Rajasthan Technical University


Jagdish Chand Bansal
South Asian University


K. V. Arya
ABV-Indian Institute of Information Techno...


Kusum Deep
Indian Institute of Technology Roorkee

# Dynamic Swarm Artificial Bee Colony Algorithm

Harish Sharma[1], Jagdish Chand Bansal[2], K. V. Arya[3], Kusum Deep[4]
[1]harish.sharma0107@gmail.com, [2]jcbansal@gmail.com, [3]kvarya@gmail.com, [4]kusumfma@iitr.ernet.in
[1,2,3]ABV-Indian Institute of Information Technology and Management, Gwalior
[4]Indian Institute of Technology, Roorkee

**Abstract:**

Artificial Bee Colony (ABC) optimization algorithm is relatively a simple and recent population based probabilistic approach for global optimization. ABC has been outperformed over some Nature Inspired Algorithms (NIAs) when tested over test problems as well as real world optimization problems. This paper presents an attempt to modify ABC to make it less susceptible to stick at local optima and computationally efficient. In the case of local convergence, addition of some external potential solutions may help the swarm to get out of the local valley and if the algorithm is taking too much time to converge then deletion of some swarm members may help to speed up the convergence. Therefore, in this paper a dynamic swarm size strategy in ABC is proposed. The proposed strategy is named as Dynamic Swarm Artificial Bee Colony algorithm (DSABC). To show the performance of DSABC, it is tested over 16 global optimization problems of different complexities and a popular real world optimization problem namely Lennard-Jones potential energy minimization problem. The simulation results show that the proposed strategies outperformed than the basic ABC and three recent variants of ABC, namely, the Gbest-Guided ABC, Best-So-Far ABC and Modified ABC.

**Keywords:** Self Adaptive Swarm; Artificial bee colony; Swarm intelligence; Dynamic Swarm; Variable Swarm.

## 1. Introduction:

Swarm Intelligence has become an emerging and interesting area in the field of nature inspired techniques that is used to solve optimization problems during the past decade. It is based on the collective behaviour of social creatures. Swarm based optimization algorithms find solution by collaborative trial and error process. Social creatures use their ability of social learning to solve complex tasks. Peer to peer learning behaviour of social colonies is the main driving force behind the development of many efficient swarm based optimization algorithms. Researchers have analyzed such behaviors and designed algorithms that can be used to solve nonlinear, non convex or combinatorial optimization problems. Previous research [8, 17, 20, 24] have shown that algorithms based on swarm intelligence have great potential to find solutions of real world optimization problems. Artificial bee colony (ABC) optimization algorithm introduced by D.Karaboga [11] is a recent addition in this category. This algorithm is inspired by the behaviour of honey bees when seeking a quality food source. Like any other population based optimization algorithm, ABC consists of a population of potential solutions. The potential solutions are food sources of honey bees. The fitness is determined in terms of the quality (nectar amount) of the food source. It is relatively a simple, fast and population based stochastic search technique in the field of nature inspired algorithms.

Since its inception, the ABC algorithm has become very popular because of its robustness and ease to apply. Many researchers have successfully applied it on the problems from different application areas. The ABC algorithm was first applied to numerical optimization problems [11]. The ABC algorithm was extended for constrained optimization problems in [15] and was applied to train neural networks [14], to medical pattern classification and clustering problems [5]. Recently Hsu et al. [10] used ABC and proposed a personalized auxiliary material recommendation system on Facebook to recommend appropriate auxiliary materials for a learner according to learning style, interests, and difficulty. The object of the proposed method was to search for suitable learning materials effectively. Fenglei et al. [18] also studied the control mechanism of local optimal solution in order to improve the global search ability of the ABC algorithm and apply it to solve TSP problems. Singh [22] used the Artificial Bee Colony algorithm for the leaf-constrained minimum spanning tree (LCMST) problem called ABC-LCMST and compared the approach against GA, ACO and tabu search (TS) [22]. Rao et al. [21] applied the ABC algorithm to network reconfiguration problem in a radial distribution system in order to minimize the real power loss, improve voltage profile and balance feeder load subject to the radial network structure in which all loads must be energized. The results obtained by the ABC algorithm were better than the other methods compared in the study, in terms of quality of the solution and computation as efficiency. Karaboga [16] used the ABC algorithm in the signal processing area for designing digital IIR filters. Pawar et al. [19] applied the ABC algorithm to some problems in mechanical engineering including multi-objective optimization of electrochemical machining process parameters, optimization of process parameters of the abrasive flow machining process and the milling process. Recently, machine intelligence and cybernetics are most well-liked field in which ABC algorithm has become a popular strategy. Bansal et. al. solved the model order reduction optimization problem of single input single output systems [4].

It has been shown that the ABC may occasionally stop proceeding towards the global optimum even though the population has not converged to a local optimum [12]. In order to overcome this problem and to speed up the convergence of ABC, a dynamic swarm artificial bee colony (DSABC) is proposed. In the proposed strategy, a dynamic swarm mechanism is integrated with the ABC. The proposed mechanism is influenced by the variable swarm strategy applied in Differential Evolution algorithm (DEVP) [25]. In the proposed strategy, the swarm size is adaptively changing through iterations based on the fitness of best-fit solution. The proposed strategy is tested over 16 well known benchmark test functions and one real world engineering optimization problem named Lennard-Jones potential energy minimization problem. To show the performance of the proposed strategy, it is compared with the basic ABC and three recent variants of ABC named Gbest-Guided ABC (GABC) [27], Best-So-Far ABC (BSFABC) [3] and Modified ABC (MABC) [1]. The simulation results show that the proposed strategy outperforms among the aforementioned algorithms.

Rest of the paper is organized as follows: section 2 describes brief overview of the basic artificial bee colony algorithm. Dynamic swarm artificial bee colony is proposed and tested in section 3. In section 4, performance of the proposed strategy is analyzed. Finally, in section 5, paper is concluded.

## 2 Artificial Bee Colony (ABC) algorithm:

In artificial bee colony algorithm, the total number of bees in the colony is divided into three groups: onlooker bees, employed bees and scout bees. Number of employed bees or onlooker bees is equal to the food sources. Employed bees are associated with food sources while onlooker bees are those bees that stay in the hive and use the information gathered from employed bees to decide the food source. Scout bee searches the new food sources randomly.

Similar to the other population-based algorithms, ABC is an iterative process which requires cycles of four phases: initialization phase, employed bees phase, onlooker bees phase and scout bee phase. Each of the phases is explained as follows:

### 2.1 Initialization of the population:

Initially, ABC generates a uniformly distributed initial population of $Np$ solutions where each solution $x_i$ $(i = 1,2,\ldots,Np)$ is a $D$-dimensional vector. Here $D$ is the number of variables in the optimization problem and $x_i$ represent the $i^{th}$ food source in the population. Each food source is generated as follows:

$$x_{ij} = x_{\min j} + rand[0,1](x_{\max j} - x_{\min j}) \qquad (1)$$

here $x_{\min j}$ and $x_{\max j}$are bounds of $x_i$ in jth direction and *rand*[0,1] is a uniformly distributed random number in the range [0, 1].

### 2.2 Employed bee phase:

In the employed bee phase, employed bees modify the current solution based on the information of individual experience and the fitness value of the new solution (nectar amount).If the fitness value of the new source is higher than that of the old source, the bee updates her position with the new one and discards the old one. The position update equation for $i^{th}$ candidate in this phase is:

$$v_{ij} = x_{ij} + \varphi_{ij} (x_{ij} - x_{ij}) \qquad (2)$$

Here, $k \in \{1,2,\ldots,Np\}$ and $j \in \{1,2,\ldots,D\}$ are randomly chosen indices. k must be different from *i*. $\varphi_{ij}$ is a random number between [-1, 1].

### 2.3 Onlooker bee phase:

After completion of the employed bee phase, the onlooker bee phase starts. In the onlooker bee phase, all the employed bees share the new fitness information (nectar) of the new solutions (food sources) and their position information with the onlooker bees in the hive. Onlooker bees analyze the available information and select a solution with a probability $prob_i$ related to its fitness. The probability $prob_i$ may be calculated using following expression (there may be some other but must be a function of fitness):

$$prob_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i} \quad (3)$$

here fitness is the fitness value of the solution i . As in the case of the employed bee, it produces a modification on the position in its memory and checks the fitness of the candidate source. If the fitness is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

### 2.4 Scout bee phase:

If the position of a food source is not updated up to predetermined number of cycles, then the food source is assumed to be abandoned and scout bees phase starts. In this phase the bee associated with the abandoned food source becomes scout bee and the food source is replaced by a randomly chosen food source within the search space. In ABC, predetermined number of cycles is a crucial control parameter which is called *limit* for abandonment. Assume that the abandoned source is $x_i$. The scout bee replaces this food source by a randomly chosen food source which is generated as follows:

$$x_{ij} = x_{\min j} + rand[0,1](x_{\max j} - x_{\min j}), \; for \; j \in \{1,2,...,D\} \quad (4)$$

where $x_{\min j}$ and $x_{\max j}$ are bounds of $x_i$ in $j^{th}$ direction.

### 2.5 Main steps of the ABC algorithm:

It is clear from the above discussion that there are three control parameters in ABC search process: The number of food sources $N_p$ (equal to number of onlooker or employed bees), the value of *limit* and the maximum number of cycles/iterations *MCN*.

In the ABC algorithm, the exploitation process is carried out by onlooker and employed bees and exploration process is carried out by scout bees in the search space. The pseudo-code of the ABC is shown in Algorithm 1 [12].

***Algorithm 1: Artificial Bee Colony Algorithm***

*Initialize control parameters and solutions $x_i (i = 1,2,...,Np)$ by using equation (1);*
*cycle = 1;*
***While** cycle <> MCN do*

- *Produce new solutions $v_i$ for the employed bees using equation (2) and evaluate them;*
- *Apply the greedy selection process for the employed bees;*
- *Calculate the probability values $prob_i$ for the solutions $x_i$;*
- *Produce the new solutions $v_i$ for the onlookers from the solutions $x_i$ selected depending on $prob_i$ using equation (3) and evaluate them;*
- *Apply the greedy selection process for the onlookers;*
- *Determine the abandoned solution for the scout, if exists, and replace it by a new randomly produced solution $x_i$ using equation (4);*
- *Memorize the best solution achieved so far;*

*cycle = cycle + 1;*
***End while***

## 3 Dynamic Swarm Artificial Bee Colony Algorithm:

Dervis Karaboga and Bahriye Akay compared the different variants of ABC for global optimization and found that the ABC shows a poor performance and remains inefficient in exploring the search space [12]. The exploration and exploitation are two important phenomena for the population-based optimization algorithms to find the global optima. Exploration of the large area of search space and exploitation of the near optimal solution region may be balanced by maintaining the diversity in early and later iterations for any random number based search algorithm. Increasing the swarm size in swarm based algorithms is one of the way which can enhance the exploration capability of the algorithm. According to newly reported results [7, 9], a reasonable choice of the swarm size in ABC is 20 to 40. For the computationally complex problems, higher swarm size based algorithms can produce better results and solve the problem of premature convergence whereas for the relatively easy problems, low swarm size based algorithms can produce early results, i.e., can show fast convergence. Therefore, it can be stated that adaptively varying swarm size through iterations according to the problem complexity can enhance the reliability (in terms of success rate) and efficiency (in terms of function evaluations)of the algorithm. Therefore, in this paper, a dynamic swarm mechanism (DSM) is proposed and incorporated with ABC. The proposed strategy is based on the variable swarm mechanism applied in Differential Evolution algorithm (DEVP) [25].

In the proposed strategy, the swarm size is set by the DSM through the progress of the search process based on status of the best solution found so far. If the fitness value of the global best solution does not improve in a predefined number of iterations (say m), then assume that the swarm is stucked to the local minima. So, it is the time to add new solutions to the current swarm to enhance its diversity. This process prohibits the search process to be stagnated. If the fitness value of the global best solution improves once or more in m iterations, then assume that the current swarm size is large enough to find better candidate solutions. To accelerate the convergence speed, the swarm size can be reduced by deleting the worst solution from the current swarm. The DSM strategy is described as follows:

– If the global best solution is not updated in m iterations then a new candidate solution is added in the swarm which is generated as follows:

$$X_{new} = X_{gbest} + \varnothing_{ij}(X_{gbest} - X_{random}), \quad (5)$$

here $X_{gbest}$ is the global best solution, $X_{random}$ is a randomly selected solution within current swarm and $X_{new}$ is the newly generated solution. If the current swarm size ($N_p$) is less than the maximum allowable swarm size (say Max$N_p$), then the newly generated solution $X_{new}$ is added in the swarm and swarm size is increased by one i.e. $N_p = N_p + 1$. If the current swarm size ($N_p$) is equivalent to Max$N_p$ then a better solution between worst of the current swarm or newly generated solution is kept in the swarm.

– If the fitness value of global best solution is updated once or more than once in $m$ iterations and if the current swarm size is greater than the minimum allowable swarm size (say Min$N_p$) then the worst solution ( say X*) in the swarm size is removed and the swarm size is reduced by one, i.e., $N_p = N_p - 1$. The pseudo-code of the proposed strategy is explained in Algorithm 2.

---

**Algorithm 2:** Dynamic Swarm Artificial Bee Colony Algorithm:

---

- Input optimization function *Minf(x)* with all decision variable restrictions;
- Initialize the solutions, $x_i(i = 1,2,...,Np)$ by using equation (1);
- Input MaxFEs, Min$N_p$, Max$N_p$ and $m$;

**while** Fes ≤ MaxFEs **do**
    **for** i =1 to $N_p$ **do**
        • Apply the basic ABC process as explained in Algorithm 1;
    **end for**
    **if** $f(X_{gbest})$ is not improved in $m$ iterations **then**
        • Generate a new solution $X_{new}$ using equation (5);
        **if** $N_p$ < Max$N_p$ then
            • $N_p = N_p + 1$;
            • $X_{Np} = X_{new}$;
        **else**
            • Select a worst solution X* in the current swarm;
            **If** $f(X_{new}) < f(X^*)$ **then**
                • Replace the worst solution in the swarm by the $X_{new}$ i.e. $X^* = X_{new}$;
            **end if**
        **end if**
    **end if**
    **if** $X_{gbest}$ is improved once or more than once in $m$ iterations **then**
        **if** $N_p$ > Min$N_p$ **then**
            • Remove the worst solution $X^*$ in current swarm;
            • $N_p = N_p - 1$;
        **end if**
    **end if**
**end while**

---

## 4 Experimental results and discussion:

### 4.1 Test problems under consideration

In order to analyze the performance of DSABC, 16 different global optimization problems ($f_1$ to $f_{16}$) are selected (listed in Table 1). These are continuous optimization problems and have different degrees of complexity and multimodality. Test problems $f_1$ to $f_6$ and $f_{13}$ *to* $f_{16}$ are taken from [2] and test problems $f_7$ *to* $f_{12}$ are taken from [23] with the associated offset.

***Table 1:*** *Test problems*

| Test Problem | Objective function | Search range | Optimum value | D | Acceptable error |
|---|---|---|---|---|---|
| Ellipsoidal | $f_1(x) = \sum_{i=1}^{D}(x_i - i^2)$ | [-D, D] | *f(1,2,3,…….D)=0* | 30 | *1.0E-05* |
| Beale function | $f_2$ (x)=[1.5-$x_1$(1-$x_2$)]$^2$+[2.25-$x_1$(1-$x_2^2$) +[2.625 - $x_1$(1-$x_2^3$)]$^2$ | [-4.5, 4.5] | *f(3, 0.5)=0* | 2 | *1.0E-05* |
| Colville function | $f_3$ (x)= 100($x_2$-$x_1^2$)$^2$ +(1-$x_1$)$^2$+90 ($x_4$-$x_3^2$)$^2$ +(1-$x_3$)$^2$ + 10.1[($x_2$-1)$^2$+($x_4$-1)$^2$]+19.8($x_2$-1)($x_4$-1) | [-10, 10] | *f(1)=0* | 4 | *1.0E-05* |
| Braninss Function | $f_4$ (x)=a($x_2$-b$x_1^2$+c$x_1$-d)$^2$+e(1-f)cos$x_1$ +e | -5≤ $x_1$≤10 0≤ $x_2$≤15 | *f(-π, 12.275)= 0.3979* | 2 | *1.0E-05* |
| Kowalik Function | $f_5$ (x) = $\sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$ | [-5, 5] | *f(0.192833,0.190836 ,0.123117,0.135766)= 0.000307486* | 4 | *1.0E-05* |
| 2D Tripod function | $f_6$ (x)=p($x_2$)(1+p($x_1$))+\|($x_1$+50p($x_2$) (1-2p($x_1$)))\|+\|$x_2$+50 (1-2p($x_2$))\| | [-100, 100] | *f(0, -50)=0* | 2 | *1.0E-04* |
| Shifted Rosenbrock | $f_7$ (x)=$\sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias}$, z=x-o+1, x=[$x_1, x_2, \dots\dots x_D$], O=[$O_1, O_2, \dots\dots O_D$], | [-100, 100] | *f(0)=*$f_{bias}$*= 390* | 10 | *1.0E-01* |

| | | | | | |
|---|---|---|---|---|---|
| Shifted sphere | $f_8(x)=\sum_{i=1}^{D} z_i^2 + f_{bias}$, z=x-o, x=$[x_1, x_2, \ldots\ldots x_D]$, O=$[O_1, O_2, \ldots\ldots O_D]$, | [-100, 100] | *$f(0)=f_{bias}=-450$* | 10 | *1.0E-05* |
| Shifted Rastrigin | $f_9(x)=\sum_{i=1}^{D}(z_i^2 - 10\cos(2\pi z_i) + 10) + f_{bias}$, z=x-o, x=$[x_1, x_2, \ldots\ldots x_D]$, O=$[O_1, O_2, \ldots\ldots O_D]$, | [-5, 5] | *$f(0)=f_{bias}=-330$* | 10 | *1.0E-02* |
| Shifted Schwefel | $f_{10}(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2 + f_{bias}$ , z=x-o, x=$[x_1, x_2, \ldots\ldots x_D]$, O=$[O_1, O_2, \ldots\ldots O_D]$, | [-100, 100] | *$f(0)=f_{bias}=-450$* | 10 | *1.0E-05* |
| Shifted Griewank | $f_{11}(x)=\sum_{i=1}^{D}\frac{z_I^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right)+1+f_{bias}$, z=x-o, x=$[x_1, x_2, \ldots\ldots x_D]$, O=$[O_1, O_2, \ldots\ldots O_D]$, | [-600, 600] | *$f(0)=f_{bias}=-180$* | 10 | *1.0E-05* |
| Shifted Ackley | $f_{12}(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi z_i))+20 + e + f_{bias}$, <br> z=x-o, x=$[x_1, x_2, \ldots\ldots x_D]$, O=$[O_1, O_2, \ldots\ldots O_D]$, | [-32, 32] | *$f(0)=f_{bias}=-140$* | 10 | *1.0E-05* |
| Six-hump camel Back | $f_{13}(x)=(4-2.1x_1^2+\frac{x_1^4}{3})x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$ | [-5, 5] | *f(-0.0898, 0.7126) =-1.0316* | 2 | *1.0E-05* |
| Easom's function | $f_{14}(x)=-\cos x_1 \cos x_2\, e^{(-(x_1-\pi)^2-(x_2-\pi)^2)}$ | [-10, 10] | *$f(\pi, \pi)=-1$* | 2 | *1.0E-13* |
| Mayer and Roth Problem | $f_{15}(x)=\sum_{i=1}^{5}\left(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i} - y_i\right)^2$ | [-10, 10] | *f(3.13,15.16,0.78)=* $0.4x10^{-4}$ | 3 | *1.0E-03* |
| Shubert Problem | $f_{16}(x)=-\sum_{i=1}^{5} i cos((i+1)x_1+1) + \sum_{i=1}^{5} i cos((i+1)x_2+1)$ | [-10, 10] | *f(7.0835, 4.8580) =-186.7309* | 2 | *1.0E-05* |

### 4.2 Real world problem:

To see the robustness of the proposed strategy, one real world global optimization problem namely Lennard Jones [6] is solved. It is a potential energy minimization problem. Lennard-Jones potential problem involves the minimization of molecular potential energy associated with pure Lennard-Jones cluster [6]. The function to minimize is a kind of potential energy of a set of $N$ atoms. The position $X_i$ of the atom $i$ has three coordinates, and therefore the dimension of the search space is $3N$. In practice, the coordinates of a point $X$ are the concatenation of the ones of the $X_i$. In short, we can write x = $[x_1, x_2, \ldots \ldots . x_N]$, and we have then

$$f_{17}(X) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\frac{1}{||X_i - X_j||^{2\alpha}} - \frac{1}{||X_i - X_j||^{\alpha}}\right)$$

In the study $N = 5$, $\alpha = 6$, and the search space is [-2, 2] [6].

### 4.3 Experimental setting:

To prove the efficiency of DSABC, it is compared with the basic ABC, GABC [27], BSFABC [3] and MABC [1]. To test DSABC and the basic ABC over test problems and solve Lennard-Jones potential problem, following experimental setting is adopted**:**

**Parameter setting for the basic ABC:**

– Colony size $CS$ =40 [7, 9],

– $\varnothing_{ij}$ = $rand$ [−1, 1],

– Number of food sources $N_p = CS/2$,

– $limit$ =1500 [13, 1],

– The stopping criteria is either maximum number of function evaluations which is set to be 200000 is reached or the acceptable error (mentioned in Table 1) have been achieved,

**Parameter setting for DSABC:** These parameter settings are based on the empirical experiments.

– $N_p$= 50

– $m$= 20, Max$N_p$=100 and Min$N_p$=10;

Remaining parameters are same as for the basic ABC. Parameter setting for the algorithms GABC, BSFABC and MABC are same as suggested by the respective authors. To illustrate the DSM mechanism, Figure 1 presents the changes of swarm size achieved by DSABC over function $f_5$, $f_{11}$, $f_{12}$ and $f_{14}$ through iterations.

### 4.4 Comparison of DSABC with ABC, GABC, BSFABC and MABC:

Numerical results with experimental setting of subsection 4.3 are given in Table 2. In Table 2, standard deviation (SD), mean error (ME), average function evaluations (AFE), and success rate (SR) are reported. Table 2 shows that the most of the time DSABC outperforms in terms of reliability, efficiency and accuracy compared to the basic ABC and its variants (GABC, BSFABC and MABC).

**Table 2:** Comparison of the results of test problems

| Test Problem | Algorithm | SD | ME | AFE | SR |
|---|---|---|---|---|---|
| $f_1$ | DSABC | 1.72E-06 | 8.18E-06 | 18570.16667 | 30 |
| | ABC | 2.43E-06 | 7.93E-06 | 20452 | 30 |
| | GABC | 2.28E-06 | 7.17E-06 | 21185.33333 | 30 |
| | BSFABC | 2.60E-06 | 6.91E-06 | 33954.66667 | 30 |
| | MABC | 7.43E-07 | 9.14E-06 | 31937.33333 | 30 |
| $f_2$ | DSABC | 1.26E-06 | 5.37E-06 | 80449.3 | 30 |
| | ABC | 7.01E-06 | 9.36E-06 | 13583.23333 | 30 |
| | GABC | 3.28E-06 | 5.00E-06 | 19584.7 | 30 |
| | BSFABC | 9.13E-06 | 3.58E-05 | 76112.3 | 25 |
| | MABC | 2.71E-06 | 5.79E-06 | 2195.066667 | 30 |
| $f_3$ | DSABC | 2.30E-03 | 1.91E-03 | 146650.0333 | 16 |
| | ABC | 1.18E-01 | 1.74E-01 | 200016.4667 | 0 |
| | GABC | 3.20E-02 | 3.96E-02 | 200020.1667 | 0 |
| | BSFABC | 2.22E-02 | 2.18E-02 | 189096.4333 | 3 |
| | MABC | 1.04E-04 | 9.52E-04 | 103253.7667 | 30 |
| $f_4$ | DSABC | 5.79E-06 | 4.82E-06 | 15308.2 | 28 |
| | ABC | 7.75E-06 | 6.74E-06 | 34963.73333 | 25 |
| | GABC | 6.53E-06 | 6.47E-06 | 27929.03333 | 26 |
| | BSFABC | 5.85E-06 | 5.56E-06 | 1744.4 | 30 |
| | MABC | 6.25E-06 | 5.11E-06 | 21202.7 | 26 |
| $f_5$ | DSABC | 6.17E-05 | 1.09E-04 | 100724.6333 | 22 |
| | ABC | 7.68E-05 | 1.93E-04 | 177490.8 | 5 |
| | GABC | 4.61E-05 | 1.09E-04 | 155300.2667 | 17 |
| | BSFABC | 6.00E-05 | 1.27E-04 | 127698.6 | 19 |
| | MABC | 1.26E-04 | 2.33E-04 | 161705.1 | 12 |

| | | | | | |
|---|---|---|---|---|---|
| $f_6$ | DSABC | 2.07E-05 | 6.95E-05 | 6078.366667 | 30 |
| | ABC | 2.12E-05 | 6.44E-05 | 15533.1 | 30 |
| | GABC | 2.49E-05 | 6.31E-05 | 9430.8 | 30 |
| | BSFABC | 7.23E-04 | 1.89E-04 | 12375.6 | 29 |
| | MABC | 2.39E-05 | 6.70E-05 | 20452.66667 | 29 |
| $f_7$ | DSABC | 6.93E-01 | 1.30E+01 | 88123.23 | 26 |
| | ABC | 1.06E+00 | 6.43E-01 | 183828.7 | 5 |
| | GABC | 2.10E-02 | 7.91E-02 | 76823.9 | 30 |
| | BSFABC | 9.28E+00 | 5.15E+00 | 171747.6 | 8 |
| | MABC | 1.12E+00 | 4.82E-01 | 181356.067 | 17 |
| $f_8$ | DSABC | 2.37E-06 | 6.60E-06 | 10123.57 | 30 |
| | ABC | 2.30E-06 | 6.91E-06 | 7452 | 30 |
| | GABC | 2.96E-06 | 6.85E-06 | 6964 | 30 |
| | BSFABC | 2.71E-06 | 7.12E-06 | 14569.33 | 30 |
| | MABC | 1.70E-06 | 8.05E-06 | 6873.33333 | 30 |
| $f_9$ | DSABC | 8.47E+00 | 8.66E+01 | 200090.2 | 0 |
| | ABC | 9.29E+00 | 8.48E+01 | 200032.4 | 0 |
| | GABC | 8.96E+00 | 8.52E+01 | 200025.4 | 0 |
| | BSFABC | 1.69E+01 | 1.23E+02 | 200023.9 | 0 |
| | MABC | 1.22E+01 | 7.81E+01 | 200014.9 | 0 |
| $f_{10}$ | DSABC | 2.25E+03 | 1.34E+04 | 200075.8 | 0 |
| | ABC | 2.99E+03 | 1.12E+04 | 200018.8 | 0 |
| | GABC | 2.26E+03 | 9.46E+03 | 200025.3 | 0 |
| | BSFABC | 7.88E+03 | 2.85E+04 | 200033.2 | 0 |
| | MABC | 2.92E+03 | 9.28E+03 | 200014.867 | 0 |
| $f_{11}$ | DSABC | 6.40E-03 | 3.87E-03 | 123611.9 | 20 |
| | ABC | 2.49E-03 | 8.27E-04 | 81740.83 | 27 |
| | GABC | 1.84E-03 | 4.98E-04 | 92379.03 | 28 |
| | BSFABC | 6.67E-03 | 5.34E-03 | 105570.5 | 17 |
| | MABC | 8.38E-03 | 6.98E-03 | 153249.967 | 13 |

| | | | | | |
|---|---|---|---|---|---|
| $f_{12}$ | DSABC | 1.85E-06 | 7.90E-06 | 19506.23 | 30 |
| | ABC | 2.10E-06 | 7.30E-06 | 13616 | 30 |
| | GABC | 1.29E-06 | 8.57E-06 | 12230.67 | 30 |
| | BSFABC | 1.66E-06 | 8.26E-06 | 25326.67 | 30 |
| | MABC | 9.54E-07 | 8.82E-06 | 10580.2667 | 30 |
| $f_{13}$ | DSABC | 1.40E-05 | 1.67E-05 | 107189.6 | 14 |
| | ABC | 1.55E-05 | 1.78E-05 | 113688.7 | 13 |
| | GABC | 1.42E-05 | 1.53E-05 | 87103.67 | 17 |
| | BSFABC | 1.53E-05 | 1.61E-05 | 93739.37 | 16 |
| | MABC | 1.48E-05 | 1.40E-05 | 80535.1 | 14 |
| $f_{14}$ | DSABC | 2.83E-14 | 4.82E-14 | 39121.03 | 30 |
| | ABC | 1.46E-04 | 8.95E-05 | 200023.2 | 0 |
| | GABC | 7.46E-05 | 2.96E-05 | 200016.6 | 0 |
| | BSFABC | 2.66E-14 | 4.23E-14 | 4030.233 | 30 |
| | MABC | 2.99E-14 | 4.00E-14 | 32324.5667 | 29 |
| $f_{15}$ | DSABC | 2.73E-06 | 1.94E-03 | 14991.83 | 30 |
| | ABC | 3.12E-06 | 1.95E-03 | 29698.07 | 30 |
| | GABC | 2.32E-06 | 1.94E-03 | 4912 | 30 |
| | BSFABC | 3.24E-06 | 1.95E-03 | 13447.83 | 30 |
| | MABC | 2.71E-06 | 1.95E-03 | 4010.16667 | 30 |
| $f_{16}$ | DSABC | 4.44E-06 | 3.88E-06 | 5677.9 | 30 |
| | ABC | 5.72E-06 | 4.89E-06 | 5531.733 | 30 |
| | GABC | 5.26E-06 | 4.75E-06 | 4247.767 | 30 |
| | BSFABC | 5.26E-06 | 4.69E-06 | 7494.967 | 30 |
| | MABC | 5.14E-06 | 4.34E-06 | 12706.8333 | 30 |
| $f_{17}$ | DSABC | 1.58E-04 | 8.40E-04 | 37768.1 | 30 |
| | ABC | 1.37E-04 | 9.11E-04 | 77503.77 | 29 |
| | GABC | 1.70E-03 | 2.24E-03 | 164158.9 | 11 |
| | BSFABC | 1.34E-04 | 8.75E-04 | 136583.5 | 29 |
| | MABC | 4.21 E-01 | 2.03E+00 | 200019.767 | 0 |

**Table 3:** Results of the comparison

| Test problems | ABC | GABC | BSFABC | MABC |
|---|---|---|---|---|
| $f_1$ | + | + | + | + |
| $f_2$ | - | - | - | - |
| $f_3$ | + | + | + | - |
| $f_4$ | + | + | - | + |
| $f_5$ | + | + | + | + |
| $f_6$ | + | + | + | + |
| $f_7$ | + | - | + | + |
| $f_8$ | - | - | + | - |
| $f_9$ | + | + | + | + |
| $f_{10}$ | + | + | + | + |
| $f_{11}$ | - | - | + | + |
| $f_{12}$ | - | - | + | - |
| $f_{13}$ | + | + | + | + |
| $f_{14}$ | + | + | - | + |
| $f_{15}$ | + | - | - | - |
| $f_{16}$ | + | - | + | + |
| $f_{17}$ | + | + | + | + |

The basic ABC, GABC, BSFABC and MABC are compared with DSABC through SR, SD and AFE. First SR is compared for all these algorithms and if it is not possible to distinguish the algorithms based on SR then comparison is made on the basis of AFE. SD is used for comparison if it is not possible on the basis of SR and AFE both. On the basis of this comparison, results of Table 2 are analyzed and Table 3 is constructed. In Table 3, '+' indicates that the DSABC is significantly better and '-' indicates that there is no significance difference or the DSABC is not comparatively better. It is observed from Table 3 that the DSABC outperforms for 48 comparisons out of the 68 comparison.

For the purpose of comparison in terms of consolidated performance, boxplot analysis is carried out for all the considered algorithms. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [26]. The boxplots are shown in Figure 2. It is clear from Figure 2 that DSABC is better than among the considered algorithms as interquartile range and median are low for DSABC.

## 5 Conclusions:

This paper is an attempt to propose a dynamic swarm in ABC optimization algorithm. Due to dynamic setting of swarm in ABC, it is more reliable and efficient than ABC. With the help of numerical experiments and their statistical analyses, DSABC is shown to be superior over basic ABC and three recent variants of ABC namely GABC, BSFABC and MABC.

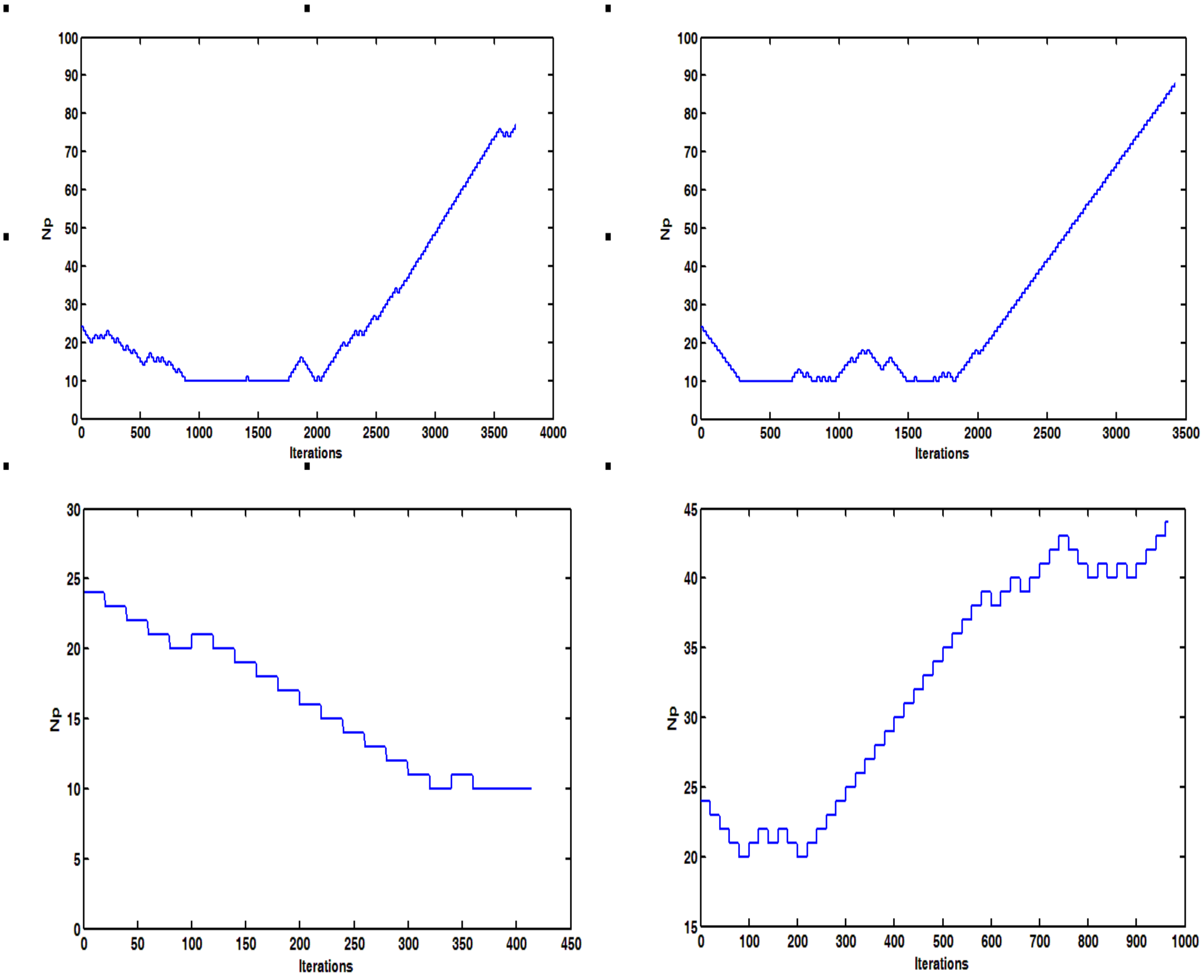


Fig. 1: Variation in the swarm size ($N_p$) through iterations; (a) for $f_5$, (b) for $f_{11}$, (c) for $f_{12}$ and (d) for $f_{14}$.

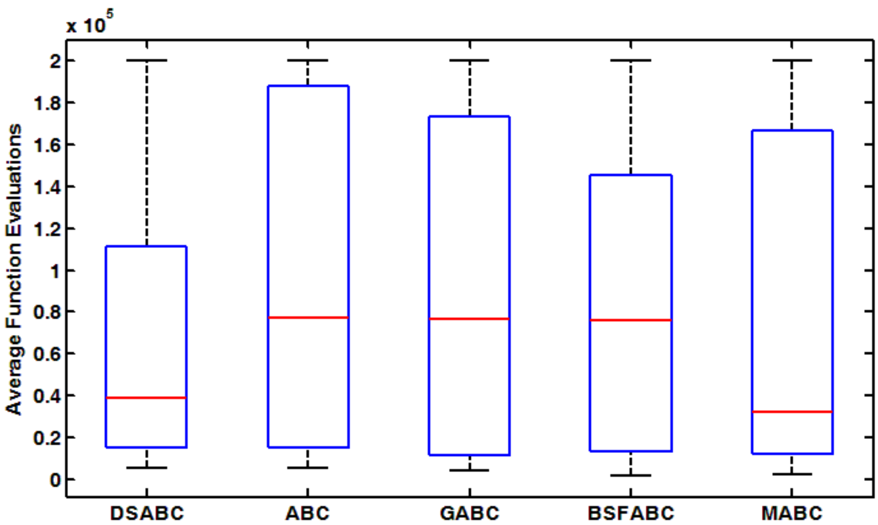


Fig. 2: Boxplots graphs for average function evaluation

## References:


1. B. Akay and D. Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. Information Sciences, doi:10.1016/j.ins.2010.07.015, 2010.

2. M.M. Ali, C. Khompatraporn, and Z.B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. Journal of Global Optimization, 31(4):635–672, 2005.

3. A. Banharnsakun, T. Achalakul, and B. Sirinaovakul. The best-so-far selection in artificial bee colony algorithm. Applied Soft Computing, 11(2):2888–2901, 2011.

4. J. Bansal, H. Sharma, and K. Arya. Model order reduction of single input single output systems using artificial bee colony optimization algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2011), pages 85–100, 2012.

5. B. Akay D. Karaboga, C. Ozturk. Training neural networks with abc optimization algorithm onmedical pattern classification. International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining (Kayseri, TURKEY), June 1923, 2008.

6. S. Das and PN Suganthan. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Kolkata,India, and Nangyang Technological University, Singapore, Tech. Rep, 2010.

7. K. Diwold, A. Aderhold, A. Scheidler, and M. Middendorf. Performance evaluation of artificial bee colony optimization and new selection schemes. Memetic Computing, pages 1–14, 2011.

8. M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, volume 2. IEEE, 1999.

9. M. El-Abd. Performance assessment of foraging algorithms vs. evolutionary algorithms. Information Sciences, 182(1):243–263, 2011.

10. C.C. Hsu, H.C. Chen, K.K. Huang, and Y.M. Huang. A personalized auxiliary material recommendation system based on learning style on facebook applying an artificial bee colony algorithm. Computers & Mathematics with Applications, 2012: doi:10.1016/j.camwa.2012.03.098.

11. D. Karaboga. An idea based on honey bee swarm for numerical optimization. Techn. Rep. TR06,Erciyes Univ. Press, Erciyes, 2005.

12. D. Karaboga and B. Akay. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 214(1):108–132, 2009.

13. D. Karaboga and B. Akay. A modified artificial bee colony (ABC) algorithm for constrained optimization problems. Applied Soft Computing, pages 789–798, 2010.

14. D. Karaboga, B. Akay, and C. Ozturk. Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks. Modeling decisions for artificial intelligence, pages 318–329, 2007.

15. D. Karaboga and B. Basturk. Artificial bee colony (abc) optimization algorithmfor solving constrained optimization problems. Foundations of Fuzzy Logic and Soft Computing, pages 789–798, 2007.

16. N. Karaboga. A new design method based on artificial bee colony algorithm for digital iir filters. Journal of the Franklin Institute, 346(4):328–348, 2009.

17. J. Kennedy and R. Eberhart. Particle swarm optimization. In Neural Networks, 1995. Proceedings., IEEE International Conference on, volume 4, pages 1942–1948. IEEE, 1995.

18. F. Xing L. Fenglei, D. Haijun. The parameter improvement of bee colony algorithm in TSP problem.Science Paper Online, 2:11–12, November 2007.

19. P. Pawar, R. Rao, and R. Shankar. Multi-objective optimization of electro-chemical machining process parameters using artificial bee colony (abc) algorithm. Advances in mechanical engineering (AME-2008), Surat, India, 2008.

20. K.V. Price, R.M. Storn, and J.A. Lampinen. Differential evolution: a practical approach to global optimization. Springer Verlag, 2005.

21. R.S. Rao, SVL Narasimham, and M. Ramalingaraju. Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm. International Journal of Electrical Power and Energy Systems Engineering, 1(2):116–122, 2008.

22. A. Singh. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem.
Applied Soft Computing, 9(2):625–631, 2009.

23. P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, YP Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. In CEC 2005,
2005.