International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No.17, December 2018 – www.ijais.org

ABC-T: Modified Artificial Bee Colony Algorithm with Parameter Tuning for Continuous Function Optimization

Sadman Sakib
Department of Computer Science & Engineering
Southeast University
Dhaka-1213, Bangladesh

Mahzabeen Emu
Department of Computer Science & Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

ABSTRACT
This paper carries a comparative study on a population-based swarm intelligence (SI) algorithm and improved modified version of that algorithm. For optimization problems, the nature-inspired algorithm works better than other algorithms. There are different types of swarm intelligence algorithms available for this purpose. Among these swarm intelligence algorithms, ABC algorithm is one algorithm where 3 types of bees are seen, employed bees, onlooker bees, scout bees. Employed bees and scout bee are responsible for exploration whereas onlooker bees are responsible for exploitation. A modified version of ABC (Artificial Bee Colony) has been implemented and then compared with the standard ABC algorithm. The comparisons are conducted on an experimental set of eleven benchmark functions. The modified version of ABC that is proposed is named ABC with tuning (ABC-T). In our analysis, the rate of exploitation and exploration was changed by maintaining one static and five dynamic ratios of employed and onlooker bees to find out which combination performs well and which combination does not perform notably. The results produced by ABC-T with different ratio of exploration and exploitation is also compared to each other to find out which combination performs better for which type of function.

Keywords
Swarm intelligence algorithm; Artificial Bee Colony algorithm; Exploitation; Exploration; Unimodal function.

1. INTRODUCTION
Real-world optimization problems are often very challenging to solve sometimes. To solve such problems, optimization mechanisms can be used, though there is no guarantee that the optimal solution can be achieved [1]. However, nature-inspired algorithms can obtain an optimal solution. The two most influential and triumphant classes or directions in bio-inspired optimization algorithms involves Evolutionary Algorithms (EAs) and Swarm Intelligence algorithms (SIAs) which are inspired by the natural evolution and collective behavior in animals respectively [2], [5].

One of the techniques to perform optimization is population-based method. Swarm intelligence algorithms are one of the kinds of the population-based method along with the evolutionary algorithm. Swarm intelligence (SI) is the collective behavior of self-organized systems, decentralized, natural or artificial. In SI the candidate solution is swarm member which can be achieved by self-organization, positive/negative feedback, fluctuation etc. [11]. SI is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. In particular, the discipline focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment. Few mentionable examples of swarm intelligence algorithms are ant colony optimization [3], particle swarm optimization [2], [5], cuckoo search [6], Artificial bee colony algorithm (ABC) [11], [12], bat algorithm [7], Firefly algorithm [9] etc.

Artificial bee colony algorithm (ABC) is a meta-heuristic algorithm which simulates the foraging behavior of honey bees. It can be used to solve various optimization problems. ABC maintains a distributed population and to handle Exploration vs. Exploitation. The swarm bees are classified into three groups: employed bees, onlooker bees and scout bees. Half of the population consists of employed bees are responsible for exploration which brings randomness into the population and the other half is onlooker bee dedicated to exploitation. Employed bees search for food sources (i.e. candidate solutions) by exploring the search space and share the information with onlooker bees through 'Waggle Dance'. The job of onlooker bee is to choose the best food source with the maximum fitness value. Scout bees search for new food sources when the previous food source is no longer beneficial. So, in the scout bee phase, exhausted food sources are being abandoned [8], [10].

Here the aim was to introduce an improved and modified version of ABC named ABC-T by changing or tuning some controlling parameters in order to get a more feasible solution. Exploration and exploitation rate is changed by maintaining some ratios to see which ratio produces the best result for which types of optimizing problems. Our numerical results show that the proposed ABC-T algorithm produces better or equal best and average solutions in all cases.

2. THE ARTIFICIAL BEE COLONY (ABC) ALGORITHM
ABC algorithm is based on the intelligent food foraging behavior of honey bees. This model was developed from the foraging behavior of a honeybee colony [10], which identifies three essential components (food sources, employed bees and unemployed bees) and two modes of behavior (recruitment to a food source and abandonment of a source).

2.1 Employed Bee Stage
This stage imitates the foraging by the employed bees in the neighborhood of their current food source positions. The position of each food source gets updated in this step. Suppose, an employed bee is currently positioned at a food source position x_i.The employed bee searches in the vicinity of x_i by producing a new trial food position v_i around x_i using equation (1).

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{kj} - x_{ij}) \tag{1}$$

Here, $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, SN\}$ are randomly picked indices.

D = the number of search dimensions

SN = the number of employed bees (or, food positions)

φ_{ij} = a uniform random value from [-1, 1].

So, the new test solution v_i is found from x_i by perturbing one of its randomly picked parameters (i.e., x_{ij}) and using the information of another randomly picked candidate solution x_k. If v_i has higher 'fitness' value than the original solution x_i, then x_i is rejected and substituted by v_i. For the problem of function optimization, F is the function that needs to be minimized. The fitness value can be calculated using the equation (2). [1]

$$fitness(x) = \begin{cases} \frac{1}{1+F(x)}, & if\ F(x) \geq 0 \\ 1 + |F(x)|, & Otherwise \end{cases} \quad (2)$$

2.2 Onlooker Bee Stage

In the onlooker bee stage, employed bee is selected randomly to follow by each onlooker bee. Suppose, p_i is the probability that an onlooker bee could select the employed bee with food source position x_i, which is computed by ABC using equation (3).

$$p_i = \frac{fitness(x_i)}{\sum_{n=1}^{SN} fitness(x_n)} \quad (3)$$

So, the probability p_i to be proportional to $fitness(x_i)$, ensuring that the probability of picking a food source is kept proportional to its quality. Similar to the employed bees, each onlooker bee also employs (1) to produce a trial food source v_i in the vicinity of its current food position x_i. If v_i has better fitness value than the old food position x_i, then x_i is replaced by v_i. Otherwise, x_i is retained and v_i is discarded. [1]

2.3 Scout Bee Stage

If a particular food source position x_i has not been enriched over an unusually long period of time (i.e., last *limit* cycles), then it is presumed to be stuck at a locally optimal point. If this happens, the ABC algorithm abandons x_i and the bee employed to x_i now becomes a scout bee that is placed at random across the search space using equation (4). [1]

Here $j = 1, 2, \ldots, D$

$[min_j, max_j]$ = the search space along the j^{th} dimension.

$$x_{ij} = min_j + rand(0,1) * (max_j - min_j) \quad (4)$$

3. PROPOSED ABC WITH PARAMETER TUNING (ABC-T)

To avoid the pitfalls of the standard ABC algorithm, an improved version of the ABC algorithm is proposed by controlling some of the parameters in a way that it is able to produce better results than the standard one. Our ABC with tuning (ABC-T) algorithm has been tuned in such a way that it changes the exploration & exploitation ratio over the generations. Also, the ABC-T algorithm allows scout bees only to a certain period of the algorithm. This upgraded ABC-T algorithm has been implemented and tested on standard benchmark problems and the performance was compared to the performance of the standard ABC algorithm. Our numerical results show that the proposed ABC-T algorithm produces better or equal best and average solutions for all cases. To maintain a balanced exploration to exploitation ratio over the course of generations is the prime objective of our proposed algorithm. In essence, all these diverse mimicking tries to accomplish two things: to exploit good found solutions (exploitation), but also go to unknown places (exploration) in order to avoid being trapped in local minima. The successfulness of any such nature-inspired algorithm is determined by a proper balance between exploitation and exploration. This balance is maintained by adjusting certain parameters and also by applying some rules in certain situations. The original algorithm maintains a 50%-50% ratio of onlooker bees & employed bees. The algorithm was tuned by changing the proportion of employed & onlooker bees over the period of running time. The number of onlooker bees increases & the number of employed bees decreases over the execution time of the algorithm. In the following table, we have our different parameters for exploration and exploitation ratio.

Table 1. ABC-T parameter change ratio

Type of parameter change	Onlooker-Employed bees
Dynamic	Linearly increasing onlooker bees & decreasing employed bees
Static	20% - 80%
	40% - 60%
	50% - 50%
	60% - 80%
	80% - 20%

As the method doesn't really need exploration at last little iteration so the scout bees were totally ignored as it introduces the exploration which was not wanted at the later part of the execution. So basically the scout bees phase work only for certain first generations. The percentage ratio was found for which algorithm derives better result. The proposed ABC-T pseudo-code is given below:

1. Initialize a pool of SN food source positions (candidate solutions), $for\ i$ = 1, 2… S. Each xi is a vector of D parameters: xi = [$xi1$, $xi2$, … , xiD]T
2. Evaluate the fitness of each food source position by the equation (2).

Repeat

3. For each employed bee, produce a new food source position vi by using equation (1).
4. Evaluate each new solution vi by using equation (2). If vi has higher fitness than the fitness of the previous xi, then accept vi to replace xi. Otherwise, discard. Set employed bee = employed bee - 10% * employed bee.
5. Then, calculate the probability value pi linked with each food position xi using the equation (2).
6. In terms of the onlooker bee, assign it to a food source position, proportionally based on the probability pi.
7. In terms of the onlooker bee, produce a new food source position vi by using equation (1).
8. Evaluate each new solution vi using equation (2). If vi has higher fitness value than xi, then accept vi to replace. Otherwise, discard. Set onlooker bee = onlooker bee + 10% * onlooker bee.
9. Check if C<= 0.2 * MCN. If yes then check if a food source has not improved during the last limit cycles, then abandon it and replace it with a new randomly placed scout bee with its food source xi produced by using equation (4).

10. Memorize the best food source position found.
11. Set cycle counter $C = C + 1$.
12. Continue until C = Maximum number of cycles (MNC)

4. EXPERIMENTAL STUDIES

4.1 Benchmark Functions

For this experimental purpose eleven benchmark functions have been used which are shown in Table 2. Among these eleven functions first three functions are unimodal functions and others are multi-modal functions.

Table 2. Benchmark Function

No	Function Name	Formulation	f_{min}	Category
f_1	*High Conditioned Elliptic Function*	$\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}}x^2$	0	Unimodal functions
f_2	*Cigar*	$x_1^2+10^6\sum_{i=2}^{D}x^2$	0	
f_3	*Discus*	$10^6x_1^2+\sum_{i=2}^{D}x^2$	0	
f_4	*Rosenbrock's*	$\sum_{i=1}^{n}[100\,(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	0	Multi-modal functions
f_5	*Ackley's*	$-20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi\, x_i\right)+20+e$	0	
f_6	*Weierstrass*	$\sum_{i=1}^{D}(\sum_{k=0}^{k\max}[a^k 2\pi b^k(x_i+0.5))])-D\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k.0.5)]$	0	
f_7	*Griewank*	$\sum_{i=1}^{n}\frac{x^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	0	
f_8	*Rastrigin*	$\sum_{i=1}^{n}[x_i^2-10\cos(2\pi\, x_i)+10)]$	0	
f_9	*Katsuura*	$\frac{10}{D^2}\prod_{i=1}^{D}(1+i\sum_{j=1}^{32}\frac{\lvert 2^j x_i-round(2^j x_i)}{2^j})^{\frac{10}{D^{1.2}}}-\frac{10}{D^2}$	0	
f_{10}	*HappyCat*	$\left\lvert\sum_{i=1}^{D}x_i^2-D\right\rvert^{1/4}+(0.5\sum_{i=1}^{D}x_i^2+\sum_{i=1}^{D}x_i^2)/D+0.5$	0	
f_{11}	*HGBat*	$\left\lvert(\sum_{i=1}^{D}x_i^2)^2-(\sum_{i=1}^{D}x_i^2)\right\rvert^{1/2}+(0.5\sum_{i=1}^{D}x_i^2+\sum_{i=1}^{D}x_i)/D+0.5$	0	

4.2 Results of ABC

Parameters adopted for the ABC algorithm are given below:

Limit: Number of onlooker bees *Dim.

Number of onlookers: 50% of the swarm

Number of employed bees: 50% of the swarm

Number of scouts: 1

These are the control parameters of ABC algorithm [11].

From the results, after performing ABC on the benchmark

functions, it is evident that ABC has produced the best result for function 2 (cigar) and worst result for function 6 (Weierstrass). It is also found that, ABC performs better in unimodal functions that multi-modal functions. The results found by the ABC algorithm have been recorded in the table 3. The Function minimum, mean and absolute error values for all the functions are represented in Fig 1, Fig 2 and Fig 3 respectively.

Table 3. Results found by performing ABC

No	Fmin	Mean	Std. Deviation	Absolute Error	Mean Absolute Error
f_1	5.0596e-24	4.3047e-23	4.1025e-23	5.0596e-24	0.24445
f_2	**4.2424e-24**	3.3745e-23	2.1118e-23	4.2424e-24	
f_3	1.1011e-12	1.3253e-11	1.697e-11	1.1011e-12	
f_4	0.34514	0.45117	0.076719	0.34514	
f_5	1.1635e-13	2.1174e-13	7.397e-14	1.1635e-13	
f_6	1.8395	2.2481	0.2651	1.8395	
f_7	0.0017634	0.0068792	0.0041171	0.0017634	
f_8	0.00088823	0.31672	0.25608	0.00088823	
f_9	0.41221	0.56037	0.096329	0.41221	
f_{10}	0.018472	0.091035	0.026737	0.018472	
f_{11}	0.071468	0.13143	0.02652	0.071468	

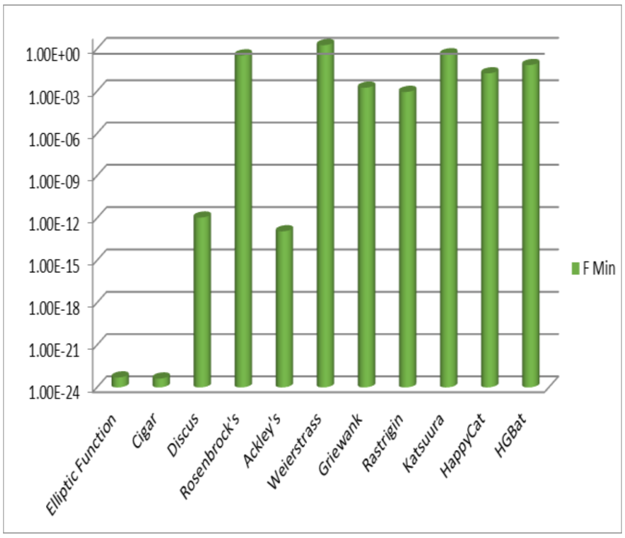

Fig 1: F-min values of ABC for all the functions

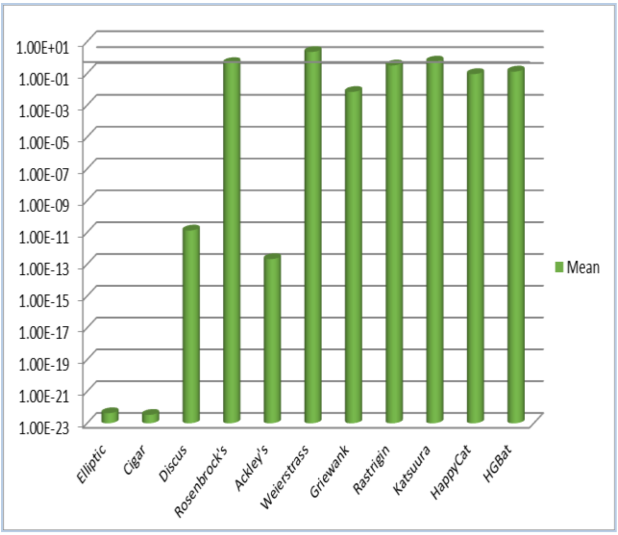

Fig 2: Mean values of ABC for all the functions

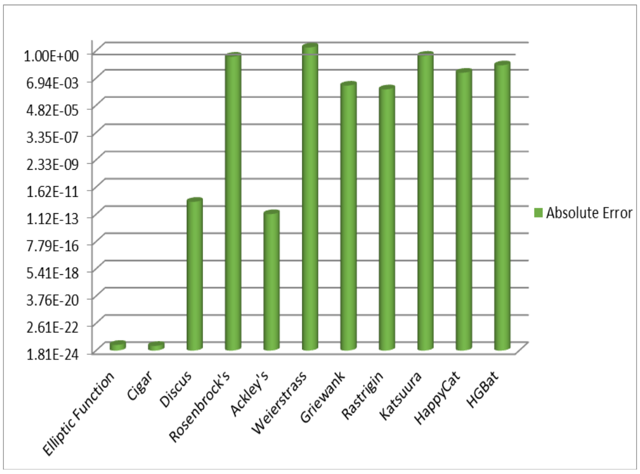

Fig 3: Absolute Error of ABC for all the functions

4.3 Results obtained from proposed ABC-T

In all the experiments, the same self-adaptive method, the same population size, the same tournament size for selection, the same initialization and the same initial population is used for ABC-T. For all functions, 20 times testing and 2000 iterations were used in each run. First, the ABC-T algorithm was run dynamically where the number of onlooker bees' decrease & the number of employed bees increase. Also for further examining, the ratio of employed bees & onlooker bees were static to record the minimum function value for each function. In the static portion, 5 different ratios were considered and one dynamic ratio mentioned in table 1, between employed and onlooker bee to examine which one performs better for which types of function.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No.17, December 2018 – www.ijais.org

Table 4. Results produced by ABC-T

No.	Type of measure	Dynamic	Static number of onlooker- employed bees in percentage				
			20%-80%	**40%-60%**	**50%-50%**	**60%-40%**	**80%-20%**
f_1	F-min	2.0034e-109	**6.3751e-118**	2.3172e-105	1.8291e-102	5.8804e-107	2.5652e-117
	Mean	4.5589e-107	**8.1677e-116**	2.326e-102	8.7141e-101	4.7594e-103	1.0506e-115
	Std. deviation	7.5941e-107	1.3422e-115	8.9239e-102	7.3034e-101	7.5017e-103	1.3617e-115
	Absolute Error	2.0034e-109	**6.3751e-118**	2.3172e-105	1.8291e-102	5.8804e-107	2.5652e-117
f_2	F-min	2.603e-109	**2.4331e-119**	2.8396e-107	7.6762e-103	3.6159e-107	2.7107e-119
	Mean	2.0049e-107	2.0065e-116	5.2266e-104	8.7141e-101	7.6339e-104	**9.4715e-117**
	Std. deviation	2.6789e-107	4.8619e-116	1.2516e-103	6.8505e-100	2.3476e-103	1.3896e-116
	Absolute Error	2.603e-109	**2.4331e-119**	2.8396e-107	7.6762e-103	3.6159e-107	2.7107e-119
f_3	F-min	5.1498e-97	9.9036e-106	4.7677e-94	3.6624e-91	1.9042e-93	**2.4063e-106**
	Mean	2.8271e-95	**1.5386e-103**	3.5085e-91	9.9083e-89	4.5785e-91	2.5097e-103
	Std. deviation	4.5321e-95	3.0015e-103	7.9056e-91	2.4118e-88	8.6244e-91	8.832e-103
	Absolute Error	5.1498e-97	9.9036e-106	4.7677e-94	3.6624e-91	1.9042e-93	**2.4063e-106**
f_4	F-min	0.012591	0.0028368	0.017027	0.032482	0.0025824	**0.00068276**
	Mean	0.15132	0.16223	0.15952	0.17166	**0.13844**	0.14849
	Std. deviation	0.052139	0.056922	0.067105	0.058032	0.078764	0.070814
	Absolute Error	0.012591	0.0028368	0.017027	0.032482	0.0025824	**0.00068276**
f_5	F-min	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**
	Mean	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**
	Std. deviation	0.26314	0	0	0	0	0
	Absolute Error	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**	**-8.8818e-16**
f_6	F-min	1.5442	**0.8089**	1.2356	1.0793	1.2245	1.5861
	Mean	2.0835	2.0289	2.0263	2.0594	2.0554	**2.1328**
	Std. deviation	0.37384	0.36462	0.35859	0.36327	0.38487	0.27032
	Absolute Error	1.5442	**0.8089**	1.2356	1.0793	1.2245	1.5861
f_7	F-min	**0**	**0**	**0**	**0**	**0**	**0**
	Mean	0.00041013	0.00017367	3.19e-06	7.7212e-05	0.00091505	**1.9176e-09**
	Std. deviation	0.0016764	0.00051036	1.0123e-05	0.00029745	0.0026755	6.0464e-09
	Absolute Error	**0**	**0**	**0**	**0**	**0**	**0**
f_8	F-min	**0**	**0**	**0**	**0**	**0**	**0**
	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	Std. deviation	**0**	**0**	**0**	**0**	**0**	**0**
	Absolute Error	**0**	**0**	**0**	**0**	**0**	**0**
f_9	F-min	0.25785	0.38037	0.3968	0.32224	**0.1993**	0.20438
	Mean	**0.48546**	0.52595	0.57464	0.53609	0.50266	0.52426
	Std. deviation	0.13151	0.090423	0.10743	0.13398	0.14561	0.16842
	Absolute Error	0.25785	0.38037	0.3968	0.32224	**0.1993**	0.20438
f_{10}	F-min	0.03332	0.046922	0.044578	**0.029902**	0.078128	0.041301
	Mean	0.075229	0.082443	0.077404	**0.071287**	0.078128	0.076099
	Std. deviation	0.022251	0.018453	0.017871	0.019518	0.01867	0.022099
	Absolute Error	0.03332	0.046922	0.044578	**0.029902**	0.078128	0.041301
f_{11}	F-min	0.03904	0.0322	0.031951	0.044784	0.076385	**0.031237**
	Mean	0.064637	0.068835	0.069771	0.074279	0.076385	**0.059413**
	Std. deviation	0.019617	0.022064	0.021858	0.018798	0.022162	0.016719
	Absolute Error	0.03904	0.0322	0.031951	0.044784	0.076385	**0.031237**
	Best F-min Count	3	**6**	3	4	4	**6**
	Best Mean Count	3	4	2	3	3	**6**
	Least Absolute Error Count	3	**6**	3	4	4	**6**

From the results it is evident that when onlooker- employed bees' percentages are 20%-80% or 80%-20% comparatively better results were shown than others. Although 80%-20% of onlooker-employed bees' ratio produced the best results among all of these methods of ABC-T. However, when controlling onlooker and employed bees dynamically the results are not quite as good.

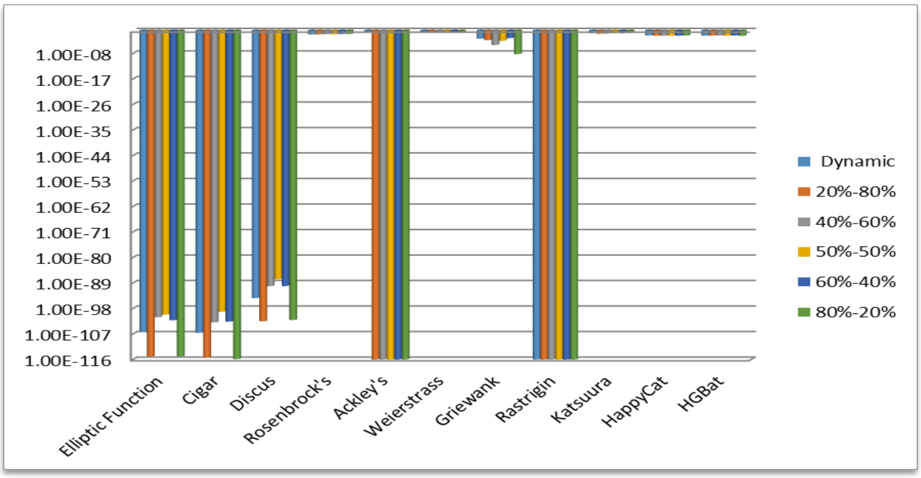

Fig 4: Function Min Values of ABC-T for all the Functions

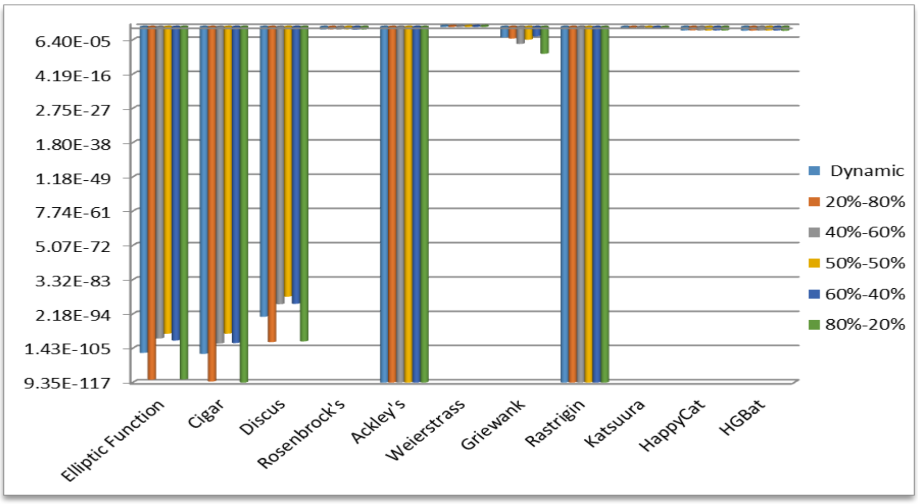

Fig 5: Mean Values of ABC-T for all the Functions

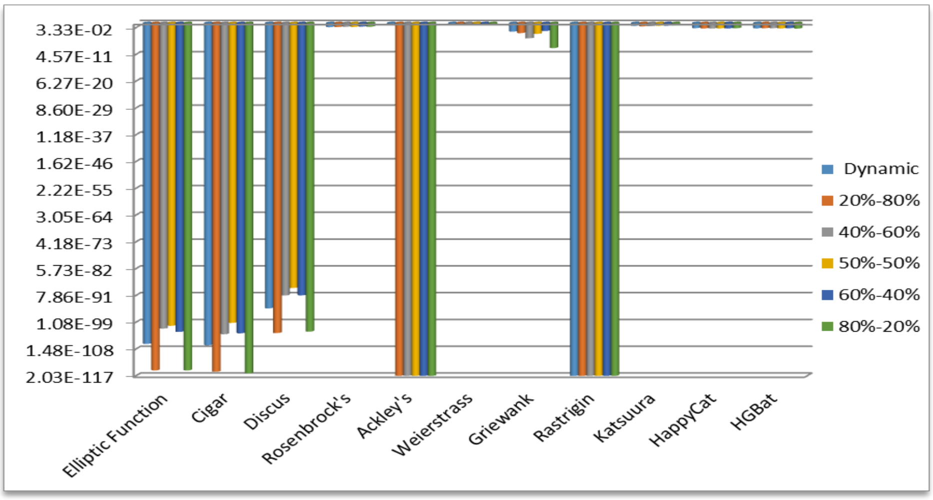

Fig 6: Absolute Error Values of ABC-T for all the Functions

5. EXPERIMENTAL RESULTS COMPARISON

From the table above it is clear that ABC-T has outperformed ABC algorithm in 10 functions among the eleven benchmark functions. This proves that our proposed modified ABC-T algorithm is an improved version of ABC algorithm for most cases.

Table 5. Comparison between ABC and ABC-T

No.	ABC	ABC-T
f_1	5.0596e-24	**6.3751e-118**
f_2	4.2424e-24	**2.4331e-119**
f_3	1.1011e-12	**2.4063e-106**
f_4	0.34514	**0.00068276**
f_5	1.1635e-13	**-8.8818e-16**
f_6	1.8395	**0.8089**
f_7	0.0017634	**0**
f_8	0.00088823	**0**
f_9	0.41221	**0.1993**
f_{10}	**0.018472**	0.029902
f_{11}	0.071468	**0.031237**
Best F min count	1	10

6. CONCLUSION

This paper presented a modified artificial bee colony algorithm with a concept mixing up the rate of exploration and exploitation. The parameters were tuned by changing the percentage of onlooker bee and employed bee. As employed bee introduces randomness or exploration the rate of employed bees were maintained higher at the beginning and kept reducing it as time goes. The proposed algorithm was tested on eleven benchmark functions. This parameter adjustment actually turned out to be a good solution as this ABC-T algorithm outperformed standard ABC algorithm according to the experimental results. This work can be proven to be useful in terms of parameter tuning and in studies related to intensification and diversification for different swarm intelligence algorithms.

7. REFERENCES

[1] I. Fister, Jr, X. Yang, I. Fister, J. Brest, D. Fister (2013), A Brief Review of Nature-Inspired Algorithms for Optimization, ELEKTROTEHNIˇSKI ESTNIK 80(3): 1–7, 2012.

[2] Binitha S, S. S. Sathya, "A Survey of Bio inspired Optimization Algorithms", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307,

[3] Dorigo, Marco & Birattari, Mauro & Stützle, Thomas.. Ant Colony Optimization. Computational Intelligence Magazine, IEEE. 1. 28-39. 10.1109/MCI.2006.329691, 2006.

[4] Das, S. Abraham, A. and Konar, A. "Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives", Springer-Verlag Berlin Heidelberg 2008.

[5] Dr. M. S. Alam, "Continuous Optimization with evolutionary and swarm intelligence algorithms", PhD Thesis, Bangladesh University of Engineering and Technology, September 2013.

[6] Civicioglu, P. and E. Besdok, "A conception comparison of the cuckoo search, particle swarm optimization, differential evolution and artificial bee colony algorithms", Artificial Intelligence Review, pp 1–32, 2011.

[7] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, 65-74, 2010.

[8] V. Tereshko, and A. Loengarov, "Collective decision-making in honeybee foraging dynamics", Computing and Information Systems Journal, Vol–9, No. 3, 2005.

[9] X.-S. Yang, "Firefly algorithms for multimodal optimization", in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178, 2009.

[10] V. Tereshko, "Reaction-diffusion model of a honeybee colony's foraging behavior", Proceedings of the 6th Parallel Problem Solving from Nature (PPSN), pp 807 816, 2000.

[11] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Erciyes University, Kayseri, Turkey, Technical Report–TR06, 2005.

[12] D. Karaboga, B. Basturk.: "A powerful and efficient algorithm for numerical function optimization: artificial Bee Colony(ABC) algorithm". J. Global Optim. 39(3), 459–471 (2007).

[13] M. S. Alam, R. Y., F. Alam, H. S. Saadi, "Experimental Comparison Between Differential Evolution and Artificial Bee Colony Algorithm: A Case Study With Continuous Optimization Problems",International Journal of Applied Information".