An Alternative Bat Algorithm

Hegazy Zaher,
Prof. of Statistics

Mahmoud El-Sherbieny,
Prof. of Operations Research

Naglaa Ragaa Saied,
Assoc. Prof. of Operations Research

Eman Oun,
Candidate of Doctoral in Operations Research

Institute of Statistical Studies & Researches, Cairo University, Egypt.

***Abstract*— This paper introduces a novel bat algorithm based on modifying the rules of the standard Bat Algorithm to solve unconstrained optimization problems. The main feature of the novel algorithm is that it arrives at the exact solution in several benchmarking problems from the first run (the mean = the best = the worst = the exact value) and in the other cases it dominates solutions in other known algorithms. A comparison is given between the results of the proposed algorithm and some well-known algorithms in the literature including the standard Bat Algorithm.**

Keywords— Evolutionary Algorithms, Bat Algorithm and Optimization Problems.

I. Introduction

Optimization problems are common in many fields. In optimization problems, the target is to find solutions, which are optimal or near optimal under given situations. The important task of optimization is to minimize wasted time or maximize the utilization of a given system [1].

Optimization algorithms are generally classified into two groups as deterministic and stochastic algorithms based on the used operators.

The deterministic algorithms use gradient information and these algorithms are ideal for functions having one global optimum, while they might have problems for functions that have several local optima. Stochastic algorithms are preferred for these functions as they can escape from local minima easily in spite of their slow convergence speed [2].

One of these algorithms is Bat algorithm (BA), proposed by Yang X. It's a swarm based meta-heuristic algorithm inspired by an echolocation property. Echolocation is a kind of sonar that aims bats in their flying and hunting behavior. Bats also could distinguish different types of insects, even in complete darkness [3].

BA is simple to understand, its control parameters are few, its convergence speed is fast, and it is simple to implement. In order to improve its performance, the researchers have applied a variety of different modifications to the standard BA. From a quick literature review, it is found a set of researches that make a set of modifications or hybridizations on the standard BA to improve its performance.

Iztok F. hybrids the differential evolution strategies and the standard BA and named it as a hybrid bat algorithm [4]. Yilmaz S. enhanced exploration and exploitation mechanisms of BA by three cumulative modifications; the first one is analyzing the structure of velocity with the inertia weighted update process. The second one is taking into consideration the difference between the solution and the global best solution to get the closest and the farthest dimension of the solution. The third modification is making hybridization between Artificial Bee Colony and BA to improve the exploration capability of BA [5].

Chen Z. removed the velocity parameter and added the inertia weight of location in BA. This inertia weight is determined using normal distribution and then the frequency of the micro bats emitted pulses adjust to the change of random position and optimal position of the micro bats [6]. Amir H. introduced chaos into BA to increase global search mobility for robust global optimization [7].

Yilmaz S. improved the exploration mechanism by equalizing the loudness A and pulse emission rate r to the problem dimension by assigning them to each dimension of the solution separately, which can perform different capabilities of exploration and exploitation simultaneously [8].

Wasi M. presented a self-adaptive BA for the global numerical optimization problems over continuous domains. It uses a selection probability to control the employing frequency, which leads to a new self-adaptive search for the BA [9]. Ali A. accelerated the process of the search using the Nelder-Mead method as a local search method [10].

Besides, of these modifications the researchers added some modifications to the standard BA to be able to handle a set of optimization applications to get better results. The structure of this paper is as follows. In Section 2, the original BA is introduced. The proposed modified algorithm is described in Section 3. Section 4 illustrates experiments and results comparison. The discussion and conclusion of this paper are presented at the end in Section 5.

II. THE STANDARD BAT ALGORITHM

As a novel property, BA depends on the echolocation characteristics of micro bats. BA depends on a frequency-tuning technique to increase the diversity of the solutions in the population, while at the same; it tries to balance exploration and exploitation during the search process by simulating the variations of pulse emission rates and loudness of bats when searching for the prey using the automatic zooming. Yang X. developed the bat algorithm with the following three idealized rules:

- All bats use echolocation to detect distance, and they distinguish the difference between food/prey;
- Bats fly randomly with velocity v_i at position x_i with a frequency f_{min}, varying wavelength λ, and loudness A_0 to search for prey. They can modify the frequency f, their loudness and the rate of pulse emission $r\epsilon\,[0{,}1]$ automatically, based on the proximity of their target;
- It assumes that the loudness varies from a large positive A_0 to a minimum constant value A_{min}.

A. Initialization of Bat Algorithm

The initial population is generated randomly for n number of bats. Each individual of the population consists of real-valued vectors with d dimensions. The equation used to generate the initial population is as follows.

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

Where $i = 1, 2, \dots, n; j = 1, 2, \dots, d;$ $x_{max\,j}$ and $x_{min\,j}$ are the upper and lower boundaries for dimension j.

B. Solution, Frequency & Velocity

In simulations, it uses virtual bats naturally among all bats. The rules for getting the positions x_i^t and velocities v_i^t in a d-dimensional search space are updated at each time step t based on the following equations

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \qquad (2)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i, \qquad (3)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (4)$$

Where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. x^* is the current global best solution which is located after comparing all the solutions among all the n bats. While $\lambda_i f_i$ is the velocity increment, it uses f_i to adjust the velocity while fixing the other factor λ_i. The range of f_i changes from a problem to another based on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is derived uniformly from$[f_{min}, f_{max}]$. When a solution is selected from the current best solutions, a new solution for each bat is generated locally using a random walk.

$$x_{new} = x_{old} + \varepsilon A^t, \qquad (5)$$

Where $\varepsilon \in [-1,1]$ is a random number, while A^t is the average loudness of all the bats at this time step.

C. Loudness and Pulse Updating

The loudness A_i and the rate r_i of pulse emission automatically update as the iterations proceed. The loudness decreases once a bat has found its prey, while the rate of pulse emission increases. The loudness can be chosen as any value of convenience. When the loudness reaches the minimum A_{min} , it means that the bat found the prey and stop emitting any sound. Loudness A_i and rate r_i are updated through the following equations.

$$A_i^{t+1} = \alpha A_i^t, \qquad (6)$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \qquad (7)$$

Where α and Υ are constants. For any $0 < \alpha < 1$ and $\Upsilon > 0$ such that $A_i^t \to 0,\ r_i^t \to r_i^0,$ as $t \to \infty$

The choice of parameters requires many experiments. Each bat should have different values of loudness and pulse emission rate. This could be achieved randomly. The loudness and emission rates are updated only if the new solutions are improved that means these bats walk towards better solution. The standard BA has many advantages; one of them is that it gets quick convergence at initial stages by switching from exploration to exploitation. This makes it an efficient algorithm when a quick solution is needed. In order to improve the performance, many modifications have been added to increase the diversity of the solution and to enhance the performance of the standard Bat algorithm as mentioned previously [11].

Algorithm I. Pseudo code of the BA [6]**.**

1. Objective function: $f(x), x = (x_1, \dots, x_d)$
2. Initialize bat population x_i and velocity $v_i, i = 1,2,\dots,n$
3. Define frequency f_i at x_i
4. Initialize pulse emission rate r_i and loudness A_i
5. **While** (t<maximum number of iterations)
6. Generate new solutions by adjusting frequency, updating velocities and location/solutions.
7. **If** (rand > r_i)
8. Select a solution among the best solutions
9. Generate a local solution around the selected best solution
10. **End If**
11. **If** (rand< Ai and f(xi)< f(x*))
12. Accept new solutions
13. Increase r_i reduce A_i
14. **End If**
15. Ranks the bats and find current best x*
16. **End While**
17. Display results.

III. THE PROPOSED BAT ALGORITHM (PBA)

In order to improve standard BA, a modification is applied to improve exploration capability of BA. In standard BA exploration and exploitation are controlled by pulse emission rate r, and this factor increases as iteration proceeds. Exploration capability of BA is improved by inserting linear decreasing step weight factor to control convergence of the global search. In this modification, the BA speed equation is removed. The search process is only controlled by the position vector; each bat updates its location depending on the frequency and its previous location taking in consideration the iteration number. So the Eq 3 is removed and Eq 4 is controlled using step weight factor which is calculated as follw

$$w = \left(\frac{gen_{max} - t}{gen_{max}}\right)^n \qquad (8)$$

Where gen_{max} is the maximum generation number; t is the current generation value and n is the number of bats used. Eq. 4 will be updated as follow

$$x_i^t = f_i \cdot w \cdot x_i^{t-1} + (1 - f_i) \cdot x^* \qquad (9)$$

IV. Experimental Results

A. Benchmark Functions

In order to verify the proposed modification efficiency of PBA, the algorithm is tested using 10 minimization benchmark test functions with zero optimal solution in different dimensions as seen in Table I. The values of "best, worst, mean, median, standard deviation" are shown in Table II.

TABLE I: BENCHMARK TEST FUNCTIONS

Function NO.	Function Name	Dimension	Range	Formulation
1	Sphere	10,30,50	[-5.12,5.12]	$f_1(x) = \sum_{i=1}^{n} x_i^2$
2	Zakharove	10,30,50	[-5,10]	$f_2(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$
3	Sum of Different Power	10,30,50	[-1,1]	$f_3(x) = \sum_{i=1}^{n} \lvert x_i\rvert^{i+1}$
4	Dixon-Price	10,30,50	[-10,10]	$f_4(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
5	Griewangk	10,30,50	[-600,600]	$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$
6	Six Hump Camel Back	2	[-5,5]	$f_6(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
7	Ackley Function	10,30,50	[-100,100]	$f_7(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)})$
8	Rastrigin	10,30,50	[-15,15]	$f_8(x) = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$
9	Schwefel's Problem	10,30,50	[-500,500]	$f_9(x) = 418.9829 * d + \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert}$
10	Rosenbrock	10,30,50	[-15,15]	$f_{10}(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$

B. Parameter Settings

Both the PBA and other algorithms are tested with 30 independent runs for each test function. The population size (no. of Bats) is set to 50. The number of generation is set to 1000, 1500 and 2000 for dimension = 10, 30 and 50 respectively for each run. The minimum frequency value is set to 0 while the maximum value is set to 1. The implementation is done using Matlab version R2013a.

C. Comparison with BA

Comparing the results with BA it's found that the PBA is better than BA as illustrated in table II.

TABLE II: COMPARISON BETWEEN BA AND PBA ON 10 STANDARD BENCHMARK FUNCTIONS

Fun	Function Name	Dimension	Algorithm	Best	Worst	Mean	Median	SD.
f_1	Sphere	10	BA	1.52	1.00e+01	4.69	4.32	2.26
			PBA	0	0	**<u>0</u>**	0	0

		30	BA	1.28e+01	3.64e+01	2.45e+01	2.41e+01	5.97
			PBA	0	0	**0**	0	0
		50	BA	2.64e+01	6.68e+01	4.66e+01	4.63e+01	1.09e+01
			PBA	0	0	**0**	0	0
f_2	Zakharove	10	BA	3.51e+01	3.13e+02	1.30e+02	9.41e+01	8.31e+01
			PBA	0	0	**0**	0	0
		30	BA	4.70e+02	1.30e+09	4.32e+07	9.62e+02	2.37e+08
			PBA	0	0	**0**	0	0
		50	BA	1.22e+03	2.65e+03	2.19e+03	2.24e+03	3.60e+02
			PBA	0	0	**0**	0	0
f_3	Sum of Different Power	10	BA	3.95e-00	8.91e+01	3.38e+01	3.36e+01	1.89e+01
			PBA	0	0	**0**	0	0
		30	BA	4.27e+02	1.17e+03	7.57e+02	7.66e+02	2.25e+02
			PBA	0	0	**0**	0	0
		50	BA	1.21e+03	2.55e+03	2.04e+03	2.19e+03	3.62e+02
			PBA	0	0	**0**	0	0
f_4	Dixon-Price	10	BA	5.45	8.63e+02	7.90e+01	3.66e+01	1.78e+02
			PBA	0.23928	0.69439	**0.375839**	0.315945	0.162081
		30	BA	6.11e+03	6.15e+04	2.35e+04	2.03e+04	1.38e+04
			PBA	0.98674	0.99672	**0.993886**	0.9958	0.003748
		50	BA	2.53e+04	1.90e+05	9.28e+04	8.33e+04	4.71e+04
			PBA	0.99466	0.99922	**0.997531**	0.99818	0.001748

f_5	Griewangk	10	BA	6.81	4.15e+01	1.86e+01	1.46e+01	9.19
			PBA	0	0	**<u>0</u>**	0	0
		30	BA	4.61e+01	1.75e+02	8.89e+01	8.77e+01	2.46e+01
			PBA	0	0	**<u>0</u>**	0	0
		50	BA	7.35e+01	2.59e+02	1.61e+02	1.63e+02	4.18e+01
			PBA	0	0	**<u>0</u>**	0	0
f_6	Six Hump Camel Back	2	BA	-1.03156	-1.02968	-1.03093	-1.03111	5.85
			PBA	-1.0316	-0.93409	**<u>-1.01811</u>**	-1.03095	0.030289
f_7	Ackley Function	10	BA	1.10e+01	1.68e+01	1.34e+01	1.35e+01	1.40
			PBA	8.88e-16	8.88e-16	**<u>8.88e-16</u>**	8.88e-16	2.07883e-31
		30	BA	1.29e+01	1.76e+01	1.55e+01	1.55e+01	1.03
			PBA	8.88e-16	8.88e-16	**<u>8.88e-16</u>**	8.88e-16	2.07883e-31
		50	BA	1.35e+01	1.77e+01	1.57e+01	1.55E+01	9.41e-01
			PBA	8.88e-16	8.88e-16	**<u>8.88e-16</u>**	8.88e-16	2.07883e-31
f_8	Rastrigin	10	BA	8.59e+01	1.94e+02	1.27e+02	1.28e+02	2.53e+01
			PBA	0	0	**<u>0</u>**	0	0
		30	BA	4.04e+02	6.07e+02	4.87e+02	4.81e+02	5.60e+01
			PBA	0	0	**<u>0</u>**	0	0
		50	BA	7.05e+02	1.13e+03	8.89e+02	8.71e+02	9.97e+01
			PBA	0	0	**<u>0</u>**	0	0
f_9	Schwefel's Problem	10	BA	1.50e+03	2.82e+03	2.27e+03	2.32e+03	2.83e+02
			PBA	1.56e^{+03}	3.40e^{+03}	**<u>2.11e^{+03}</u>**	1.98e^{+03}	5.92e^{+02}

		30	BA	6.71e+03	1.01e+04	8.72e+03	8.96e+03	1.12e+03
			PBA	7.50e+03	9.54e+03	**8.24e+03**	8.06e+03	$6.45e^{+02}$
		50	BA	1.39e+04	2.16e+04	1.85e+04	1.97e+04	2.86e+03
			PBA	1.36e+04	1.50e+04	**1.45e+04**	1.47e+04	$5.42e^{+02}$
f_{10}	Rosenbrock	10	BA	1.50e+03	1.22e+05	4.10e+04	2.60e+04	3.66e+04
			PBA	$1.44e^{+01}$	$2.11e^{+01}$	$\mathbf{1.84e^{+01}}$	$1.82e^{+01}$	1.97
		30	BA	7.38e+04	1.38e+06	3.78e+05	3.47e+05	2.56e+05
			PBA	$5.66e^{+01}$	$6.91e^{+01}$	$\mathbf{6.18e^{+01}}$	$6.15e^{+01}$	4.32
		50	BA	1.96e+05	2.06e+06	8.14e+05	7.14e+05	4.48e+05
			PBA	$9.32e^{+01}$	$1.20e^{+02}$	$\mathbf{1.07e^{+02}}$	$1.08e^{+02}$	9.02

D. Comparison Between the PBA and Some Other Algorithms in the Literature

Comparing the results with previous modifications like Hybrid Bat Algorithm (HBA) [4], Modified Bat Algorithm (MBA) [8], Novel Adaptive Bat Algorithm (NABA) and Bat Algorithm with Self-Adaptive Mutation (BA-SAM) [9], it's found that the PBA is superior to others as illustrated in table III.

TABLE III: COMPARISON BETWEEN HBA, MBA, NABA, BA-SAM AND PBA ON 5 STANDARD BENCHMARK FUNCTIONS

Fun	Function Name	Dimension	Algorithm	Best	Worst	Mean	Median	SD.
f_1	Sphere	10	HBA	4.83e-09	2.89e-03	1.26e-04	5.66e-04	1.66e-07
			MBA	3.73e-03	1.60e-02	8.80e-03	7.74e-03	3.34e-03
			NABA	1.20e-04	1.72	2.18e-01	7.47e-02	4.20e-01
			BA-SAM	1.76e-06	3.98e-01	9.04e-02	4.64e-02	1.05e-01
			PBA	0	0	**0**	0	0
		30	HBA	3.37e-01	9.97e+01	3.09e+00	7.67e+02	1.72e+01
			MBA	1.07e-02	1.95e-01	4.61e-02	3.06e-02	4.38e-02

			NABA	1.05e-03	2.49e+01	2.44	9.18e-02	5.64
			BA-SAM	5.77e-03	3.73	2.87e-01	6.74e-02	6.85e-01
			PBA	0	0	**0**	0	0
		50	HBA	-	-	-	-	-
			MBA	5.87	2.30e+01	1.08e+01	1.02e+01	3.70
			NABA	5.51e-02	4.57e+01	6.41	3.55e-01	1.22e+01
			BA-SAM	1.36e-01	5.87	1.03	5.63e-01	1.24
			PBA	0	0	**0**	0	0
f_5	Griewangk	10	HBA	2.25e-09	3.97e-05	3.18e-06	8.66e-06	1.14e-07
			MBA	2.05	2.06e+01	8.12	6.62	5.39
			NABA	2.68	5.06e+01	1.74e+01	1.41e+01	1.21e+01
			BA-SAM	1.31e-01	1.11e+01	2.72	1.17	2.98
			PBA	0	0	**0**	0	0
		30	HBA	6.38e-06	3.57e+01	6.42e-05	5.99e+01	3.12
			MBA	6.36e+01	1.82e+02	1.10e+02	1.01e+02	2.83e+01
			NABA	4.67e+01	1.70e+02	8.56e+01	8.21e+01	2.63e+01
			BA-SAM	7.18	8.68e+01	3.38e+01	2.67e+01	2.17e+01
			PBA	0	0	**0**	0	0
		50	HBA	-	-	-	-	-
			MBA	2.16e+02	4.26e+02	3.19e+02	3.21e+02	6.19e+01
			NABA	7.73e+01	2.35e+02	1.49e+02	1.41e+02	4.04e+01
			BA-SAM	1.34e+01	1.20e+02	5.91e+01	5.21e+01	2.90e+01

			PBA	0	0	**<u>0</u>**	0	0
f_7	Ackley Function	10	HBA	6.31e-04	2.00e+01	1.16e+01	9.26	1.78e+01
			MBA	3.61e-02	1.79	1.67e-01	6.91e-02	3.60e-01
			NABA	1.20e-01	1.15e+01	3.10	4.96e-01	4.36
			BA-SAM	2.09e-02	6.44	9.93e-01	2.68e-01	1.54
			PBA	8.88e-16	8.88e-16	**<u>8.88e-16</u>**	8.88e-16	2.07883e-31
		30	HBA	5.43e-04	9.85e+01	2.53e-03	2.15e+02	1.94e+01
			MBA	7.53	1.38e+01	1.11e+01	1.10e+01	1.63
			NABA	3.19e-01	1.42e+01	6.51	4.39	5.14
			BA-SAM	1.76	1.21e+01	4.09	3.59	2.45
			PBA	8.88e-16	8.88e-16	**<u>8.88e-16</u>**	8.88e-16	2.08e-31
		50	HBA	-	-	-	-	-
			MBA	1.31e+01	1.63e+01	1.45e+01	1.45e+01	8.07e-01
			NABA	3.21	1.55e+01	1.03e+01	1.18e+01	4.01
			BA-SAM	3.85	1.08e+01	6.68	6.28	1.91
			PBA	8.88e-16	8.88e-16	**<u>8.88e-16</u>**	8.88e-16	2.08e-31
f_8	Rastrigin	10	HBA	5.12	2.38e+01	1.55e+01	4.46	1.69e+01
			MBA	1.46e+01	3.48e+01	2.49e+01	2.55e+01	4.35
			NABA	1.34e+01	5.08e+01	2.85e+01	2.68e+01	9.33
			BA-SAM	6.73	3.83e+01	2.33e+01	2.56e+01	8.85
			PBA	0	0	**<u>0</u>**	0	0
		30	HBA	1.62	3.98e+01	1.29e+01	1.26e+03	5.03

			MBA	2.72e+01	2.11e+02	1.63e+02	1.77e+02	4.40e+01
			NABA	8.00e+01	3.08e+02	1.71e+02	1.68e+02	6.15e+01
			BA-SAM	8.69e+01	2.50e+02	1.70e+02	1.71e+02	4.72e+01
			PBA	0	0	**0**	0	0
		50	HBA	-	-	-	-	-
			MBA	1.85e+02	5.58e+02	3.84e+02	4.00e+02	1.21e+02
			NABA	2.29e+02	5.20e+02	3.70e+02	3.71e+02	7.82e+01
			BA-SAM	2.02e+02	5.49e+02	3.68e+02	3.74e+02	8.47e+01
			PBA	0	0	**0**	0	0
f_{10}	Rosenbrock	10	HBA	6.34e-02	5.10e+02	6.22e+01	1.15e+02	7.73
			MBA	-	-	-	-	-
			NABA	5.87e-01	4.89e+04	2.55e+03	1.31e+01	1.00e+04
			BA-SAM	2.72	4.70e+02	3.24e+01	1.03e+01	8.75e+01
			PBA	$1.44e^{+01}$	$2.11e^{+01}$	**$1.84e^{+01}$**	$1.82e^{+01}$	1.97
		30	HBA	8.28	2.17e+02	6.59e+01	4.00e+03	2.00e+01
			MBA	-	-	-	-	-
			NABA	2.79e+01	1.88e+05	1.23e+04	7.73e+01	4.00e+04
			BA-SAM	3.88e+01	3.24e+04	2.86e+03	1.97e+02	7.86e+03
			PBA	$5.66e^{+01}$	$6.91e^{+01}$	**$6.18e^{+01}$**	$6.15e^{+01}$	4.32
		50	HBA	-	-	-	-	-
			MBA	-	-	-	-	-
			NABA	2.11e+02	4.76e+05	4.29e+04	9.66e+02	1.18e+05

			BA-SAM	3.14e+02	1.08e+06	4.12e+04	1.27e+03	1.96e+05
			PBA	$9.32e^{+01}$	$1.20e^{+02}$	$\underline{\mathbf{1.07e^{+02}}}$	$1.08e^{+02}$	9.02

The results are tested using t-test in table IV. The results show that the proposed modification is superior to others at a degree of freedom 29 and confidence intervals 95% and 99% where the value of t-tabled at α=0.05 is 1.699 and at α=0.01is 2.462.

Table IV. T-TEST BETWEEN PBA AND OTHERS

t-value				
Function Name	**Algorithm**	**10 Dimension**	**30 Dimension**	**50 Dimension**
Sphere Function	HBA	41.57**	0.98	-
	MBA	14.43**	5.76**	15.99**
	NABA	2.84**	2.37*	2.88**
	BA-SAM	4.2**	2.29*	4.55**
Griewangk Function	HBA	15.27**	1.13e-4	-
	MBA	8.25**	0.21	0.28
	NABA	7.88**	17.83**	0.2
	BA-SAM	4**	8.53**	11.16**
Ackley Function	HBA	0.04	7.14e-4	-
	MBA	2.54**	37.3**	98.41**
	NABA	3.89**	6.94**	14.07**
	BA-SAM	3.53**	9.14**	19.16**
Rastrigin Function	HBA	5.02**	14.05**	-
	MBA	31.35**	20.29**	17.38**
	NABA	16.73**	15.23**	25.92**
	BA-SAM	14.42**	19.73**	23.8**
Rosenbrock Function	HBA	30.07**	1.1	-
	MBA	-	-	-
	NABA	1.39	1.68	1.99*
	BA-SAM	0.88	1.95*	1.15

V. DISCUSSION AND CONCLUSION

A. Discussion

The results in Table II show that PBA gets the optimal solution of several benchmark functions. It clearly demonstrates that the proposed algorithm outperforms the original BA on all benchmark test functions for all the dimensionalities. It is noted that in the proposed algorithm equation 3 of the standard algorithm is removed which leads to less consumed time. Thus the exploration and convergence characteristics of proposed algorithm are much better than the standard BA and Table III shows that PBA is also better than other proposed modified algorithms.

B. Conclusion

This paper introduces a new modification on BA and it's evaluated using a number of unconstrained benchmarking problems on numeric optimization. The proposed modified algorithm has been implemented and tested on several benchmarking optimization problems. The results (i.e., both final solution quality and convergence characteristics) clearly demonstrate that the proposed algorithm is superior on the original BA. Furthermore, it is compared with a set of other modified algorithms and proved its superiority on them.

REFERENCES

[1] S. S. Rao, *Engineering Optimization*, vol. 56, no. 772801201. 2009.

[2] X. Yang, *Firefly Algorithm*. 2010.

[3] X.-S. Yang, “A New Metaheuristic Bat-Inspired Algorithm,” *Nat. Inspired Coop. Strateg. Optim. (NICSO 2010)*, pp. 65–74, 2010.

[4] I. F. Jr and X. Yang, “A Hybrid Bat Algorithm,” vol. 80, pp. 1–7, 2013.

[5] S. Yilmaz and E. U. Kucuksille, “Improved Bat Algorithm (IBA) on Continuous Optimization Problems,” *Lect. Notes Softw. Eng.*, vol. 1, no. 3, pp. 279–283, 2013.

[6] C. Zhen, Z. Yongquan, and L. U. Mindi, “A

simplified Adaptive Bat Algorithm Based on Frequency," *J. Comput. Inf. Syst.*, vol. 9: 16 (201, pp. 6451–6458, 2013.

[7] A. H. Gandomi and X. S. Yang, "Chaotic bat algorithm," *J. Comput. Sci.*, vol. 5, no. 2, pp. 224–232, 2014.

[8] S. Yilmaz, E. U. Kucuksille, and Y. Cengiz, "Modified bat algorithm," *Elektron. ir Elektrotechnika*, vol. 20, no. 2, pp. 71–78, 2014.

[9] M. Wasi, U. Kabir, and M. S. Alam, "Bat Algorithm with Self-adaptive Mutation: A Comparative Study on Numerical Optimization Problems," *Int. J. Comput. Appl.*, vol. 100, no. 10, pp. 975–8887, 2014.

[10] A. Fouad Ali, "Accelerated Bat Algorithm for Solving Integer Programming Problems.," *Egypt. Comput. Sci. J.*, vol. 39, no. 1, p. 25, 2015.

[11] X.-S. Yang and X. He, "Bat algorithm: literature review and applications," *Int. J. Bio-Inspired Comput.*, vol. 5, no. 3, pp. 141–149, 2013.