A Novel Adaptive Bat Algorithm to Control Explorations and Exploitations for Continuous Optimization Problems

Md. Wasi Ul Kabir
Ahsanullah University of Science and Technology

Nazmus Sakib
Ahsanullah University of Science and Technology

Syed Mustafizur Rahman Chowdhury
Northern University Bangladesh

Mohammad Shafiul Alam, Ph.D
Ahsanullah University of Science and Technology

ABSTRACT
Swarm intelligence (SI) algorithms generally come from nature or biological behavior of nature. These algorithms use probabilistic search methods that simulate the behavior of biological entities or the natural biological evolution. Swarm intelligence (SI) is based on collective behavior of self-organized systems. Typical swarm intelligence algorithms include Particle Swarm Optimization (PSO), Ant Colony System (ACS), Bacteria Foraging (BF), the Artificial Bee Colony (ABC), and so on. Recently some new swarm based algorithms like Firefly Algorithm (FA) and Bat Algorithm (BA) has emerged. BA is a new optimization technique, which is based on the echolocation behavior of bats. BA is very efficient in exploitations but relatively poor in explorations. In this paper, a Novel Adaptive Bat Algorithm (NABA) is presented to improve the explorative characteristics of BA. The proposed algorithm incorporates two techniques within BA to improve its degree of explorations, which include the Rechenberg's 1/5 mutation rule and the Gaussian probability distribution to produce mutation step sizes. Both these techniques try to balance between the explorative and exploitative properties of BA. Simulation results on a number of benchmark functions on the continuous optimization problem suggest that the proposed algorithm – NABA often show much improved results, compared to the standard BA.

Keywords
Optimization, Bat Algorithm, metaheuristics, swam intelligence, bio-inspired algorithm.

1. INTRODUCTION
Optimization algorithms have extensively been applied on diverse and numerous areas over the last few decades. Many natural-world optimization problems are very complicated, where better optimization algorithms are needed [1] [2]. The aim of an optimization algorithm is to find a set of values for the parameters, i.e., the independent variables that maximizes or minimizes the value of one or more dependent variables. Many complex optimization problems cannot be solved within bounded computation time. So the algorithms capable of finding near-optimal or at least practically good solutions within reasonable computation time have drawn the attention of the scientific community. During the last few decades, the scientific community has observed the emergence of a number of nature-inspired optimization algorithms. Swarm Intelligence (SI) is subfield of Computational Intelligence that is dedicated to mimic the behavior of natural swarms to find solutions for complex optimization problems which are not easily tackled by other approaches [3]. Swarm intelligence (SI) is based on collective behavior of self-organized systems. Typical swarm intelligence algorithms include Particle Swarm Optimization (PSO), Ant Colony System (ACS), Bacteria Foraging (BF), the Artificial Bee Colony (ABC), and so on [4]. Recently some new swarm based algorithms like Firefly Algorithm (FA), Harmony search (HS) and Bat Algorithm (BA) emerges in the field of optimization algorithms [5] [6]. The algorithm proposed in this paper is based on the Bat algorithm, which established on the echolocation performance of bats.

Bat Algorithm is a relatively new swarm based algorithm which is proposed by Xin-She Yang [5]. The effectiveness of echolocation of micro bats is captivating as bats may find prey [7]. Different types of insects can be identified by these micro bats even in exhaustive lightlessness. Though Bat algorithm is very good at exploitation, it lacks exploration capability as the iterations proceed. To solve this problem a new Novel Adaptive Bat Algorithm (NABA) is proposed. The objective is to search the space more successfully by improving the accuracy and the precision of the existing Algorithm. To analyze the efficiency of the proposed Novel Adaptive Bat Algorithm (NABA), it is applied on ten continuous benchmark functions (both unimodal and multimodal, low and high dimensional) and the results are compared with existing Bat Algorithm.

The paper is organized as follows. Section 2 describes the basic Bat algorithm with a detailed pseudo-code. Section 3 presents the proposed Novel Adaptive Bat Algorithm (NABA) with a brief analysis. Section 4 provides details of benchmarking problems, parameter setting of the algorithms and compares their results.

2. BAT ALGORITHM
Bat Algorithm is a metaheuristic approach that is based echolocation behavior of bats. The bat has the capability to find its prey in complete darkness [8]. This algorithm is developed on this hunting behavior of bats. Bats are mammals with wings. Bats are born with the advanced capability of echolocation [9] [1]. Microbats are insectivores. Echolocation is a special type of sonar, used by the microbats to avoid obstacles, detect prey, and pinpoint their roosting crevices in the dark [7]. Bats emit a high sound frequency to listen the echo that bounces back from the neighboring objects. Bats radiated frequency differs in specifications. The frequency is associated with their food gathering strategies. Bats use short, frequency-modulated signals to sweep through about an octave. Signal bandwidth of bats varies depends on the species [10] [11].

The echolocation characteristics of microbats emphasize some approximate or idealized rules, by which the variation of Bat Algorithm or Bat Inspired Algorithm may be developed [1].

The approximate or idealized rules of echolocation characteristics of microbats are as follows.

i. *Distance:* Bats use echolocation to sense distance. They acknowledge the ranges/spaces between prey and surrounded barriers in some miraculous ways.
ii. *Frequency:* Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission r in the range of [0, 1], depending on the proximity of their target [12].
iii. *Loudness:* Though loudness can vary in many ways. Here it is assumed that the loudness differs from a large A_0 to a minimum constant value A_{min}.

2.1 Initialization of Bat Algorithm

Initial population is generated randomly for n number of bats. Each individual of the population consists of real valued vectors with d dimensions [12]. The following equation is used to generate the initial population:

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

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

2.2 Solution, Frequency & Velocity

Step size to generate new solution in Bat Algorithm is defined by the frequency [12]. The pulse frequency is an arbitrary value for each solution, ranges between upper and lower boundaries f_{min} and f_{max}. Frequency controls the pace and range of movement and update bat position and velocity. The velocity and position of bat is updated using 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 $\beta \in [0,1]$ indicates randomly generated number, f_i is the frequency generated for solution i, V_i represents the new velocity for solution i, x^* represents the global best solutions in the population [13].

In order to increase exploration, in r (pulse rate) probability, a solution is selected among the best solution and Radom walk is applied. Thus a new candidate solution is generated.

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

$\overline{A^t}$ is average Loudness of all the Bats $\varepsilon \in [0,1]$ is random number and represents directions and intensity of random walk.

2.3 Loudness and Pulse Emission Rate

Loudness and pulse emission rate must be adjusted as iterations proceed. When the Bat gets closer to its prey the loudness A usually decreases and pulse emission rate increases. By the following equations loudness A and pulse emission rate r are updated:

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

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

Where α and γ are constants which having the determined values for these equations. Here r_i^0 and A_i are factors whose consist of random values and A_1^0 can be normally [1, 2], while r_i^0 can be normally [0, 1] [14].

The following figures show the characteristics of equation 6 and 7 as the iteration proceeds.

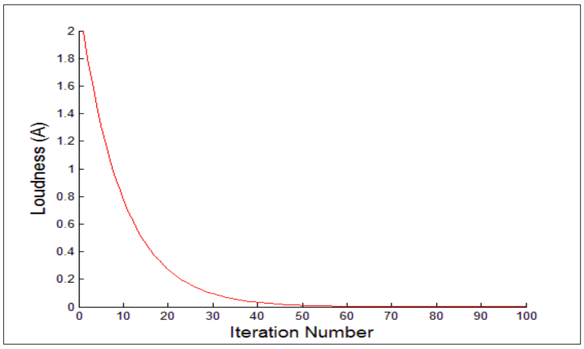

Fig 1: Loudness (A)

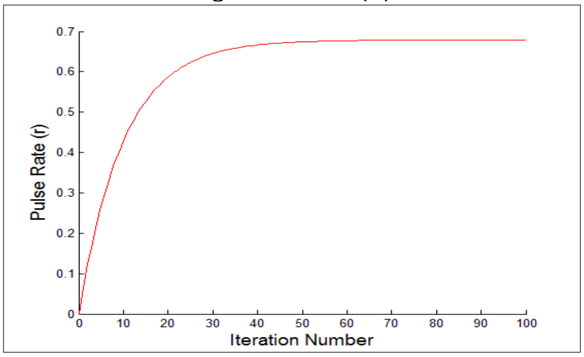

Fig.2. Pulse Emission Rate (r)

2.4 Pseudo code of the Bat Algorithm

1. Objective function: $f(x), x = (x_1, x_2, x_3, \dots \dots \dots, x_d)t$
2. Initialize bat population x_i and velocity $v_i\,; i = (1,2, \dots \dots \dots, n)$
3. Define pulse frequency f_i at x_i
4. Initialize pulse 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.
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 < A_i)and and $f(x_i) < f(x^*)$
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.

3. PROPOSED NOVEL ADAPTIVE BAT ALGORITHM

3.1 Problems in Bat Algorithm

The proposed Novel Adaptive Bat Algorithm (NABA) is developed from the existing Bat Algorithm. Bat Algorithm uses loudness A and pulse emission rate r along with the other factors (population numbers, search dimension, maximum cycle number) of population based algorithm for the optimization process. As the iterations proceed in Bat Algorithm, the loudness A decreases while the pulse emission rate r increases exponentially [15]. (Fig 1 & Fig 2)

BA algorithm loses exploration capability rapidly at the following iterations because the condition at 7th line of the algorithm is less likely to be satisfied as the iteration proceeds. Exploitation capability of BA increases when the candidate solutions are generated around the best solution. In following iterations, BA will be bad in exploration but good at exploitation and it is very likely that the algorithm will be trapped in local optima. [13] On the other hand, the process of updating solutions in equations (2), (3) and (4) increases the exploration capability of BA. In this case, BA will be very good at exploration and but bad in exploitation. So the algorithm needs a well balance between exploration and exploitation and pulse emission rate r serves for this purpose. Emission rate r increases as the iterations proceed. So the condition $rand > r_i$ is likely accepted at the beginning iterations and the algorithm performs exploitation at first steps of iterations. But at the following iterations, the possibility of $rand > r_i$ decreases and the algorithm performs exploration.

3.2 Novel Adaptive Bat Algorithm (NABA)

To eliminate the problems in basic Bat algorithm, a Novel Adaptive Bat Algorithm (NABA) is proposed in this paper. This Novel Adaptive Bat Algorithm (NABA) is focused on adaptive mutation step size and Rechenberg's 1/5 mutation rule [16]. By using these two new ideas, better result is found from the proposed Algorithm.

3.2.1 Adaptive Mutation Step Size:

In line 9 of original Bat algorithm, a new local solution is generated around the selected best solution. This solution is generated by random walk using the following formula:

$$x_{new} = x_{old} + \varepsilon \overline{A^t} \tag{8}$$

To balance between exploration and exploitation, adaptive mutation step size is used to generate new solution rather than doing just random walk.

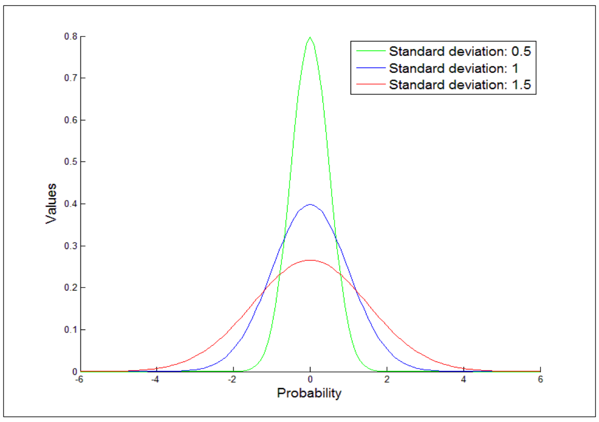

Fig. 3: Adaptive Mutation Step Size

Gaussian distribution is used to generate new solution in the new Bat algorithm. Gaussian distribution generally generates small numbers but occasionally it makes large numbers too. The distribution takes two parameters: the mean μ and standard deviation σ. The degree of small numbers over large ones can be controlled by simply changing the standard deviation σ of the distribution (Fig 3). Thus the new bat algorithm controls the random walk step size by the variance of Gaussian/Normal distribution. The modified equation is as follows:

$$x_{new} = x_{old} + \varepsilon \overline{A^t}\, N(0, \sigma) \tag{9}$$

3.2.2 Rechenberg's 1/5 mutation rule:

Rechenberg's 1/5 mutation rule is proposed by Ingo Rechenberg [17], which states that the ratio of successful mutations to all mutations should be 1/5 [18] [16]. So that:

i. If more than 1/5 mutations are successful, then the algorithm is exploiting local optima too much and standard deviation σ should be increased.
ii. If less than 1/5 mutations are successful, then the idea of algorithm is exploring too much and standard deviation σ should be decreased.
iii. If exactly 1/5 mutations are successful, then the algorithm is balanced between explorations and exploitations, so do not change the standard deviation.

Rechenberg's rule is used to adaptively change the random walk step size and pulse rate to control the exploration and exploitation. Changing the pulse rate and standard deviation according to 1/5 rule in every m number of cycles is performed as in following equations:

Pulse rate,

$$r(t+1) = \begin{cases} r(t) * 0.85 & if\ successrate(m) < 1/5 \\ r(t)\ /\ 0.85 & if\ successrate(m) > 1/5 \\ r(t) & otherwise \end{cases}$$

Standard deviation,

$$std(t+1) = \begin{cases} std(t) + 0.0001 & if\ successrate(m) < 1/5 \\ std(t) - 0.0001 & if\ successrate(m) > 1/5 \\ std(t) & otherwise \end{cases}$$

4. SIMULATION AND ANALYSIS

4.1 Benchmark functions

To evaluate the performance of the proposed algorithm — NABA, a set of ten standard benchmark functions is used. The benchmark set include unimodal, multimodal, high-dimensional and low dimensional optimization functions. A function is called multimodal if it has multiple local optima. In order to minimize such a function, the search process must be able to avoid being trapped at the regions around local minima to reach the global minima. The benchmark functions are as follows:

Table 1. Benchmark Functions used in the experimental studies. Here, *D*: Dimensionality of the function, *S*: search space, *C*: function characteristics with values — *U*: Unimodal and *M*: Multimodal.

No.	Name	*D*	*C*	*S*	Function Definition	f_{min}
f_1	Sphere	10,30,50	U	$[-5.12, 5.12]^D$	$f_1(x) = \sum_{i=1}^{d} x_i^2$	0.0
f_2	Step	10,30,50	U	$[-100, 100]^D$	$f_2(x) = \sum_{i=1}^{d}(\lfloor x_i + 0.5 \rfloor^2)$	0.0
f_3	Zakharov	10,30,50	U	$[-5, 10]^D$	$f_3(x) = \sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^4$	0.0
f_4	Griewangk	10,30,50	M	$[-600, 600]^D$	$f_4(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos\frac{x_i}{\sqrt{i}} + 1$	0.0
f_5	Rastrigin	10,30,50	M	$[-15, 15]^D$	$f_5(x) = \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	0.0
f_6	Rosenbrock	10,30,50	U	$[-15, 15]^D$	$f_6(x) = \sum_{i=1}^{d-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	0.0
f_7	Ackley	10,30,50	M	$[-32, 32]^D$	$f_7(x) = -20exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	0.0
f_8	Schwefel	10,30,50	M	$[-500, 500]^D$	$f_8(x) = 418.9829 * d - \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	0.0
f_9	Easom	10,30,50	M	$[-2\pi, 2\pi]^D$	$f_9(x) = -(-1)^d * \left(\prod_1^d cos^2(x_i)\right) * exp\left[\sum_1^d (x_i - \pi)^2\right]$	-1.000
f_{10}	Michalewicz	10,30,50	M	$[0, \pi]^D$	$f_{10}(x) = -\sum_{i=1}^{d}\sin(x_i)sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	NA

4.2 Parameter Settings for the algorithms

Algorithms are tested with 30 independent runs for each test functions. The population size (no. of Bats) is set to 50. Maximum number of generation is set to 1000, 1500 and 2000 for D = 10, 30 and 50 respectively for each run. Minimum frequency value is set to 0 while the maximum value is set to 1. Both α and λ are set to 0.9 in this simulation. The implementation of these algorithms is done using Matlab R2013a. The value of m is set to 50.

4.3 Experimental Results

In order to verify the performance of proposed modifications (NABA), both of the algorithms are tested on ten benchmark test functions with different dimensions as seen in Table 1. The values of best, worst, mean, median and standard deviation of the results found over the different runs are shown in Table 2.

Table 2. **Comparison between BA and NABA on 10 standard benchmark functions. Both the algorithms were run 30 times on each of the functions. The best result in each row is marked with boldface font.**

Fun	Name	Dim	Algorithm	Best	Worst	Mean	Median	SD
1	Sphere	10	BA	2.48E+00	2.40E+01	5.78E+00	5.06E+00	4.04E+00
			NABA	1.22E-04	4.34E-01	**7.27E-02**	3.44E-02	1.06E-01
		30	BA	1.38E+01	3.47E+01	2.44E+01	2.45E+01	5.76E+00
			NABA	1.05E-03	1.22E+01	**1.22E+00**	7.38E-02	3.07E+00
		50	BA	2.57E+01	7.44E+01	4.29E+01	4.25E+01	1.01E+01
			NABA	6.31E-02	2.62E+01	**1.69E+00**	3.40E-01	4.85E+00
2	Step	10	BA	6.60E+02	5.94E+03	1.83E+03	1.56E+03	1.04E+03
			NABA	0.00E+00	5.28E+03	**4.76E+02**	1.14E+02	1.01E+03
		30	BA	5.77E+03	1.60E+04	1.09E+04	1.13E+04	2.62E+03
			NABA	4.20E+02	1.32E+04	**4.89E+03**	4.16E+03	3.45E+03
		50	BA	8.27E+03	2.91E+04	1.68E+04	1.68E+04	5.02E+03
			NABA	3.71E+03	2.08E+04	**8.93E+03**	7.76E+03	4.08E+03

3	Zakharov	10	BA	4.01E+01	3.82E+04	1.44E+03	1.51E+02	6.95E+03
			NABA	1.11E-03	1.78E+02	**6.11E+00**	1.22E-01	3.24E+01
		30	BA	6.72E+02	7.08E+09	3.82E+08	1.16E+03	1.50E+09
			NABA	3.72E+00	7.13E+02	**8.16E+01**	1.15E+01	1.63E+02
		50	BA	1.27E+03	4.27E+11	1.89E+10	2.30E+03	8.00E+10
			NABA	7.46E+01	4.27E+11	**1.42E+10**	2.32E+02	7.79E+10
4	Griewangk	10	BA	6.77E+00	3.98E+01	2.18E+01	2.16E+01	8.11E+00
			NABA	3.93E+00	3.61E+01	**1.32E+01**	1.21E+01	6.82E+00
		30	BA	5.84E+01	1.67E+02	9.33E+01	8.76E+01	2.66E+01
			NABA	4.21E+01	1.41E+02	**7.62E+01**	7.48E+01	2.32E+01
		50	BA	8.88E+01	2.24E+02	1.54E+02	1.52E+02	3.08E+01
			NABA	7.88E+01	2.23E+02	**1.42E+02**	1.33E+02	3.81E+01
5	Rastrigin	10	BA	7.66E+01	2.12E+02	1.28E+02	1.22E+02	3.04E+01
			NABA	1.66E+01	9.16E+01	**3.72E+01**	3.04E+01	1.89E+01
		30	BA	3.85E+02	7.81E+02	4.96E+02	4.99E+02	7.87E+01
			NABA	8.78E+01	2.67E+02	**1.87E+02**	1.96E+02	4.40E+01
		50	BA	6.66E+02	1.14E+03	8.94E+02	9.04E+02	1.19E+02
			NABA	1.90E+02	7.10E+02	**3.68E+02**	3.58E+02	1.12E+02
6	Rosenbrock	10	BA	7.73E+02	1.96E+05	3.79E+04	2.02E+04	4.87E+04
			NABA	5.13E+00	2.82E+03	**1.22E+02**	1.33E+01	5.11E+02
		30	BA	1.58E+05	9.67E+05	4.03E+05	3.50E+05	1.93E+05
			NABA	3.10E+01	2.16E+05	**8.11E+03**	1.17E+02	3.94E+04
		50	BA	2.37E+05	2.54E+06	9.99E+05	8.81E+05	5.88E+05
			NABA	2.43E+02	8.16E+05	**4.82E+04**	1.95E+03	1.51E+05
7	Ackley	10	BA	1.14E+01	1.69E+01	1.34E+01	1.31E+01	1.31E+00
			NABA	5.47E-02	1.34E+01	**2.46E+00**	5.63E-01	4.01E+00
		30	BA	1.24E+01	1.71E+01	1.50E+01	1.52E+01	9.13E-01
			NABA	3.65E-01	1.57E+01	**4.89E+00**	3.14E+00	4.70E+00
		50	BA	1.42E+01	1.75E+01	1.58E+01	1.57E+01	7.97E-01
			NABA	3.90E+00	1.59E+01	**9.69E+00**	1.03E+01	3.84E+00
8	Schwefel	10	BA	2.06E+03	3.06E+03	2.63E+03	2.68E+03	2.33E+02
			NABA	1.59E+03	2.99E+03	**2.29E+03**	2.24E+03	3.48E+02
		30	BA	8.00E+03	1.07E+04	9.60E+03	9.74E+03	5.43E+02
			NABA	7.56E+03	1.04E+04	**8.98E+03**	8.86E+03	7.33E+02
		50	BA	1.22E+04	1.83E+04	1.66E+04	1.68E+04	1.37E+03
			NABA	1.27E+04	1.77E+04	**1.57E+04**	1.58E+04	9.86E+02
9	Easom	10	BA	-2.06E-04	-1.48E-52	-6.91E-06	-1.84E-12	3.75E-05
			NABA	-9.97E-01	-7.19E-39	**-6.57E-01**	-8.55E-01	4.09E-01
		30	BA	-1.65E-28	-4.12E-186	-6.14E-30	-2.61E-43	3.01E-29
			NABA	-9.70E-01	-4.12E-186	**-5.63E-01**	-8.33E-01	4.38E-01
		50	BA	-1.74E-58	-3.23E-236	-6.04E-60	-1.22E-81	3.18E-59
			NABA	-4.11E-04	0.00E+00	**-1.37E-05**	-5.84E-24	7.50E-05
10	Michalewicz	10	BA	-5.67E+00	-2.87E+00	-3.90E+00	-3.74E+00	6.75E-01
			NABA	-8.03E+00	-4.34E+00	**-5.75E+00**	-5.41E+00	1.04E+00
		30	BA	-1.08E+01	-6.96E+00	-8.52E+00	-8.65E+00	9.66E-01
			NABA	-2.05E+01	-7.34E+00	**-1.10E+01**	-9.84E+00	3.34E+00

		50	BA	-1.58E+01	-9.53E+00	-1.24E+01	-1.25E+01	1.21E+00
			NABA	-1.83E+01	-1.16E+01	**-1.39E+01**	-1.34E+01	1.88E+00

Although the performance of both the algorithms drops as the dimensionality of the functions increases, our proposed algorithm – NABA outperforms the basic BA on all benchmark test functions with all the dimensions. The search space exponentially increases with the increasing dimensions, so more effective strategies are needed for the high dimensional functions to effectively search the promising regions across the search space.

5. CONCLUSION

In this work, the performance of standard and proposed versions of the Bat Algorithm is investigated and compared their performances on the continuous optimization problems. The basic Bat Algorithm gained better results by optimizing the lower-dimensional functions but shows poor performance on the higher dimensional multimodal functions. In order to improve performance on these high dimensional problems, two techniques has been proposed in this paper — the variance of Gaussian distribution is used to tune the mutation step size and Rechenberg's rule is used to balance between explorations and exploitations adaptively. From the experimental results, it can be concluded that the Novel Adaptive Bat Algorithm has significantly outperformed the basic Bat Algorithm on the benchmark functions. In future, this initial work will hopefully be continued with more experiments, especially on the large-scale global optimization problems.

6. REFERENCES

[1] G. Wang and L. Guo, "A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization," *Journal of Applied Mathematics,* vol. 2013, p. 21, 2013.

[2] S. Yang X, "Nature-inspired Metaheuristic Algorithms," Luniver Press, 2008.

[3] R. Eberhart and J. Kennedy, Swarm Inteligence, Academic Press, 2001.

[4] j. Kennedy and R. Eberhart, "Practicle swarm optimization," in *IEEE International Conference Neural Networks*, Perth, Australia, 1995.

[5] X. S. Yang and J. R. Gonzalez, ""A New Metaheuristic Bat-Inspired Algorithm" in Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)," *Springer Press,* vol. 284, pp. 65-74, 2010.

[6] X. S. Yang, "Harmony Search as a Metaheuristic Algorithm,Music-Inspired Harmony Search Algorithm," *Theory and Applications,Studies in Computational Intelligence,* vol. 191, pp. 1-14, 2009.

[7] T. Colin, The Varienty of Life, Oxford University Press, 2000.

[8] J. Altringham, Bats: Biology and Behaviour, Oxford University Press, 1996.

[9] P. Richardson, "Bats," National history Museum , London, 2008.

[10] B. A. Faritha and C. Chandrasekar, "An optimized approach of modified bat algorithm to record deduplication," *International Journal of Computer Applications,* vol. 62, no. 1, pp. 10-15, 2012.

[11] G. komarasamy and A. Wahi, "An Optimized K-Means Clustering Technique using Bat Algoritm," *European Journal of Scientific Research,* vol. 84, no. 2, pp. 263-273, August 2012.

[12] K. Khan and A. Sahai, "A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context," *I.J. Intelligent Systems and Applications*, pp. 23-29, June 2012.

[13] E. U. K. Y. C. S. Yılmaz, "Modified Bat Algorithm," *ELEKTRONIKA IR ELEKTROTECHNIKA,* vol. 20, no. ISSN 1392-1215, p. 2, 2014.

[14] S. Yang X, "Bat algorithm for multi-objective optimization," *International Journal of Bio-Inspired Computation,* vol. 3, no. 5, pp. 267-274, 2011.

[15] Y. Selim and K. E. U. , "Improved Bat Algorithm (IBA) on Continuous Optimization Problems," *Lecture Notes on Software Engineering,* vol. 1, no. 3, pp. 279-283, August 2013.

[16] Rechenberg, Evolutionstrategie: Optimirung Technisher Systeme Nach Prinzipen des Biologischen Evolution, FrommanHozlboog, Stuttgard, Germany, 1973..

[17] I. Rechenberg, "Evolutionsstrategie," *Frommann-Holzboog*, 1994.

[18] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford, UK: Oxford University Press, 1996.