International Journal of Computer Applications (0975 – 8887)
Volume 100– No.10, August 2014

Bat Algorithm with Self-adaptive Mutation: A Comparative Study on Numerical Optimization Problems

Md. Wasi Ul Kabir
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Mohammad Shafiul Alam
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

ABSTRACT

Swarm intelligence is an emerging research field that tries to mimic the collective intelligent behavior found in swarms of insects and animals. Many algorithms have been proposed that simulate these intelligent swarm models to solve a wide range of scientific and engineering problems. The Bat algorithm is one of the most recent swarm intelligence based algorithms that simulates the intelligent hunting behavior of the bats found in nature. In this paper, we present an improved self-adaptive Bat algorithm (BA-SAM) for the problem of global numerical optimization over continuous domains. We have introduced two improved solution search equations — the BA/Normal/1 and BA/Cauchy/1 schemes. We have also used a selection probability to control the frequency of employing BA/Normal/1 and BA/Cauchy/1, which leads to a new self-adaptive search mechanism for the Bat algorithm. Experiments are conducted on both unimodal and multimodal continuous benchmark functions. The results demonstrate the improved performance of the BA-SAM algorithm in comparison to the original Bat algorithm and another recently introduced improved variant of the Bat algorithm.

Keywords

Bat algorithm, Numeric optimization, Meta-heuristic algorithms, Swarm intelligence.

1. INTRODUCTION

Over the last few decades, scientists and researchers have applied the analogy from natural and biological systems, which include the cooperative, distributed and intelligent problem solving techniques found in swarms of bees, bats, birds, ants and many others [**1**], [**2**] in order to handle the challenges and complexities of many scientific, engineering and mathematical problems. These meta-heuristic algorithms are found to be extremely effective to solve many search and optimization problems. Swarm intelligence algorithms are a group of meta-heuristic algorithms that use evolutionary steps while simulating natural swarm intelligence. Swarm intelligence uses collaborative intelligence of all the swarm members towards better society decision making [**1**]. Each member of the swarm has its small, local vision, but when properly combined, their overall vision is a big, global one, which emerges globally intelligent behavior from the local interactions, associations and self-organizations of the swarm members. Ant colony optimization (ACO), particle swarm optimization (PSO) and artificial bee colony (ABC) are a few of the common examples of many existing swarm intelligence algorithms [**2**] [**3**].

The Bat algorithm is a relatively new optimization technique [**4**]. Bats found in nature are observed to use their echo system in the process of locating their prey insects. To improve the performance of the original Bat algorithm, our objective is to experiment with several learning strategies in its mutation phase [**5**]. But finding the most suitable strategy for a specific problem may require a huge amount of computational efforts. Besides, during the different stages of evolution, different strategies with different global and local search capabilities might be preferred [**6**]. Therefore, we intend to introduce a novel Bat algorithm that can automatically adapt the learning strategies during the evolution. Some related research works on strategy adaption with evolutionary algorithms can be found in [**7**].

The rest of this paper is organized as follows. Section 2 describes the basic Bat algorithm with a detailed pseudo-code. Section 3 presents the proposed Bat Algorithm with Self-adaptive Mutation (BA-SAM) with a brief analysis. Section 4 provides details of benchmark problems, parameter settings of the different algorithms and makes a comparison among their results.

2. BAT ALGORITHM

The Bat algorithm is a recently introduced meta-heuristic algorithm for search and optimization, which is first proposed by Xin-She Yang in 2010 [**4**]. Bat Algorithm is based on the echolocation behavior of bats. Each bat has an interesting capability to find its prey in complete darkness. This algorithm is developed on this hunting behavior of bats. Bats are mammals with wings and they are born with the advanced capability of echolocation [**8**] [**9**]. Echolocation is a special type of sonar, used by the bats to avoid obstacles, detect prey, and pinpoint their location in the dark. Bats emit a high sound frequency to listen the echo that bounces back from the neighboring objects. The frequency is associated with their food gathering strategies [**10**].

The idealization of the echolocation of bats can be summarized as follows [**11**]. Bats use echolocation to sense distance. They acknowledge the ranges/spaces between prey and surrounded barriers in some miraculous ways. 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**]. It is assumed that the loudness differs from a large A_0 to a minimum constant value A_{min}. Each step of basic bat algorithm is explained below:

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. The following equation is used to generate the initial population.

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

Where $i = 1,2, \dots\dots, n\,; j = 1,2, \dots\dots, d$; The $x_{\max j}$ and $x_{min\,j}$ are the 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. The pulse frequency is an arbitrary value for each solution, ranges between upper and lower boundaries f_{min} and f_{max}. The frequency controls the pace and range of the movement and updates the bat position and velocity. The velocity and position of bat is updated using the following equations [**13**].

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

$$V_i^t = V_i^{t-1} + (x_i^t - x^*)f_i \quad (3)$$

$$x_i^t = x_i^{t-1} + V_i^t \quad (4)$$

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

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

$\overline{A^t}$ is the average Loudness of all the bats and $\varepsilon \in [0,1]$ is random number that represents the directions and intensity of random walk [**14**].

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 r increases. By the following equations loudness A and pulse emission rate r are updated [**15**]:

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

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

Where α and γ are constants and is set to 0.9 in our simulation. Here the initial loudness A_i^0 can typically be [1, 2], while the initial emission rate r_i^0 can be normally [0, 1] [**10**]. The following figures show the characteristics of equation (6) and (7) as the iteration proceeds.

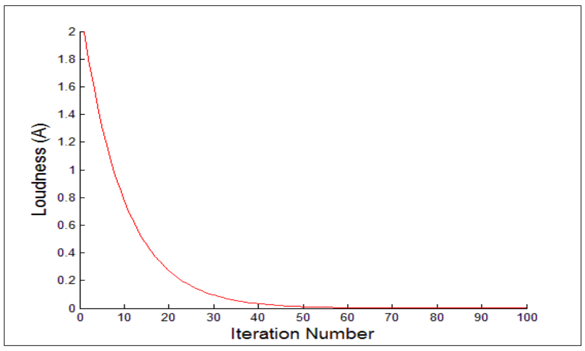

Fig. 1: Loudness (A)

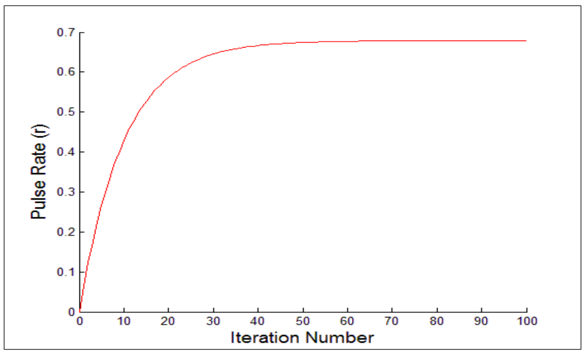

Fig. 2: Pulse Emission Rate (r)

2.4 Pseudo code of the Bat algorithm

```
1. Objective function: $f(x),\ x = (x_1, x_2, x_3, \dots\dots, x_d)^T$
2. Initialize bat population $x_i$ and velocity $v_i\,; i =$
   $(1,2, \dots\dots, n)$
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,
7.    and updating velocities and locations/solutions
8.     If (rand > $r_i$)
9.        Select a solution among the best solutions
10.       Generate a local solution around the selected best
           solution
11.    end if
12.    If (rand < $A_i$) and and $f(x_i) < f(x^*)$
13.        Accept new solutions
14.        Increase $r_i$, reduce $A_i$
15.    end if
16.    Ranks the bats and find current best $x^*$
17. end while
18. Display results.
```

3. PROPOSED BAT ALGORITHM WITH SELF-ADAPTIVE MUTATION (BA-SAM)

3.1 Mutation using Cauchy and Normal distributions:

Probability distributions are very important in theory and applications of different field. There are literally infinitely many probability distributions. Among them, Gaussian/Normal, Cauchy, Levy distributions are very widely used. Gaussian distribution generally generates small numbers but occasionally it makes large numbers too [**13**]. Gaussian 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 [**7**]. On the other hand, Cauchy distribution is the distribution of a random variable that is the ratio of two independent standard normal variables and has the following probability density function.

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2} \quad (8)$$

Where *t>0* is a scale parameter. The corresponding distribution function is:

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{t}\right) \quad (9)$$

The shape of $f_t(x)$ is almost the same as the Gaussian density function but approaches the axis so slowly that an expectation does not exist [**13**] [**7**]. So the variance of the Cauchy distribution is infinite. Fig. 3 shows the difference between Cauchy and Gaussian functions by plotting them in the same scale. Due to its long flat tails (Fig. 3), the Cauchy mutation is more likely to generate an offspring further away from its parent than the Gaussian mutation [**7**]. It is expected to have a higher probability of escaping from a locally optimal point or moving away from a plateau, which makes the Cauchy mutation more explorative than the Gaussian mutation.

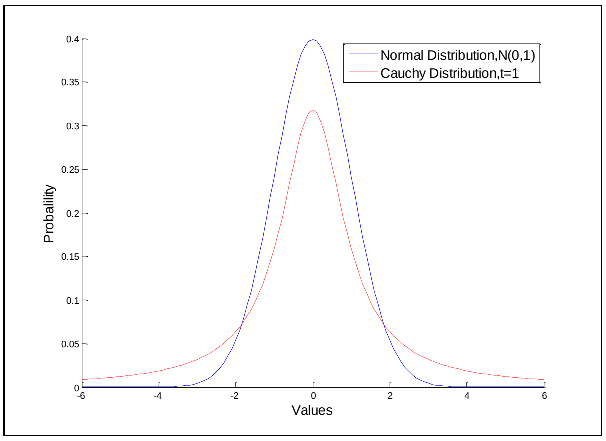

Fig. 3: Comparison between Cauchy and Gaussian density functions.

In basic Bat algorithm, a new local solution is generated around the selected best solution by random walk using the following equation [**4**].

$$x_{new} = x_{old} + \varepsilon\overline{A^t} \tag{10}$$

In our proposed algorithm BA-SAM, we have tried to balance between explorations and exploitations by producing the mutation step size using both Gaussian and Cauchy distributions. This is significantly different from the original Bat algorithm which generates new solutions from just random walks. The modified equations for BA-SAM are as follows.

$$\text{"BA/Normal/1"}: x_{new} = x_{old} + \varepsilon\overline{A^t}\,N(0,1) \tag{11}$$

$$\text{"BA/Cauchy/1"}: x_{new} = x_{old} + \varepsilon\overline{A^t}\,C(0,1) \tag{12}$$

3.2 Selection probability among the two strategies:

The central idea for self-adaptation in the proposed algorithm — BA-SAM is to probabilistically select one out of its several available mutation schemes by using a learning strategy. We have two mutation schemes available here to produce the candidate solutions — the BA/Normal/ and BA/Cauchy/1. The BA/Normal/1 strategy is supposed to show exploitative properties and good convergence characteristics, while the BA/Cauchy/1 strategy is more explorative and should preserve more population diversity that is necessary to avoid premature convergence around the locally optimal points. Let the probability of applying the strategy BA/Normal/1 to each individual in the current population is p_1 while the probability of applying the other strategy is p_2 = 1-p_1. Initially both the strategies have equal probability (p_1= p_2=0.5) to be applied to each individual of the initial population.

We randomly generate a real valued vector of size *n* (Population size) with uniform distribution in the range [0, 1] for each individual. If the *k*-th element value of the vector is smaller than or equal to p_1, then the strategy

BA/Normal/1 will be applied on the *k*-th individual. Otherwise, the strategy BA/Cauchy/1 will be applied. After each generation, the number of individuals successfully entering the next generation by the strategies BA/Normal/1 and BA/Cauchy/1 are recorded as nos_1 and nos_2, respectively. The number of individuals discarded after being generated by the BA/Normal/1 and BA/Cauchy/1 are recorded as nof_1 and nof_2. The total number of success and failure are accumulated within a specified number of generations *L*, which is called the learning period. We used *L* = 50, and after each *L* generations, the probability of p_1 and p_2 are updated as follows.

$$p_1 = \frac{nos_1\,(nos_2 + nof_2)}{nos_1(nos_2 + nof_2) + nos_2(nos_1 + nof_1)} \tag{13}$$

$$p_2 = 1 - p_1 \tag{14}$$

After each learning period, the probability of applying both the strategies is updated again and again. The counters nos_1, nos_2, nof_1 and nof_2 are reset after each update in order to avoid the possible side-effect accumulated from the previous learning period.

4. SIMULATION AND ANALYSIS

4.1 Benchmark functions

In this experiment, we have evaluated the new proposed algorithm — BA-SAM using a set of 10 benchmark problems on numeric optimization. Among the benchmark functions, the f_1–f_5 are unimodal functions, while f_6–f_{10} are multimodal functions. For the multimodal functions, the number of local minima increases exponentially with the problem size, i.e., the number of search dimensions. Functions are tested with the dimensions *d* = 10, 30 and 50. The search space, global minimum, dimensionality, characteristics and analytical form of each function are shown in Table 1.

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	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_3	Step	10,30,50	*U*	$[-100, 100]^D$	$f_4(x) = \sum_{i=1}^{d}(\lfloor x_i + 0.5\rfloor^2)$	0.0
f_4	Quartic	10,30,50	*U*	$[-1.28, 1.28]^D$	$f_5(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	0.0
f_5	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_6	Schwefel	10,30,50	M	$[-500, 500]^D$	$f_2(x) = 418.9829 * d - \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	0.0
f_7	Rastrigin	10,30,50	M	$[-15, 15]^D$	$f_7(x) = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$	0.0
f_8	Griewangk	10,30,50	M	$[-600, 600]^D$	$f_8(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\frac{x_i}{\sqrt{i}} + 1$	0.0
f_9	Ackley	10,30,50	M	$[-32, 32]^D$	$f_9(x) = -20exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i\right) + 20 + e$	0.0
f_{10}	Penalized	10,30,50	M	$[-50, 50]^D$	$f_{10}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4), y_z = 1 + \frac{1}{4}(x_i + 1)$ $u(x_{i,}a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	0.0

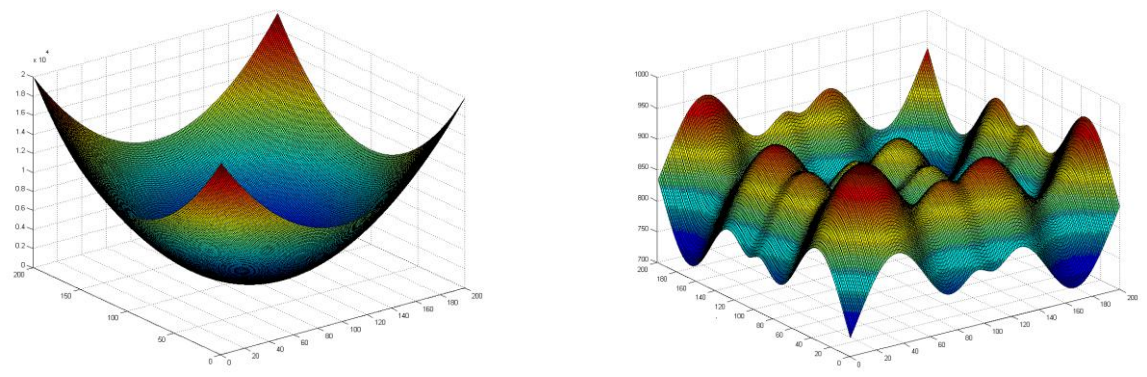

Fig. 4: 3D surface plots of 2D Sphere function (left) and Schwefel function (right).

4.2 Parameter Settings for the algorithms

Both the 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 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 value of learning period L is set to 50. The implementation of these algorithms is done using Matlab version R2013a.

4.3 Experimental Results

In order to compare the performance of the proposed algorithm BA-SAM, the algorithms are tested on ten benchmark test functions with different dimensionalities as seen in Table 1. We compare this new algorithm with our previous work – Novel Adaptive Bat Algorithm (NABA) and the basic Bat algorithm [**13**]. The values of the 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, NABA and BA-SAM on 10 standard benchmark functions. All three algorithms are run 30 different times on each of the functions. The best result for each function with each dimensionality is marked with boldface font.**

Fun	*Name*	*Dim*	*Algorithm*	*Best*	*Worst*	*Mean*	*Median*	*SD*
f_1	Sphere	10	BA	1.52E+00	1.00E+01	4.69E+00	4.32E+00	2.26E+00
			NABA	1.20E-04	1.72E+00	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
		30	BA	1.28E+01	3.64E+01	2.45E+01	2.41E+01	5.97E+00
			NABA	1.05E-03	2.49E+01	2.44E+00	9.18E-02	5.64E+00
			BA-SAM	5.77E-03	3.73E+00	**2.87E-01**	6.74E-02	6.85E-01
		50	BA	2.64E+01	6.68E+01	4.66E+01	4.63E+01	1.09E+01
			NABA	5.51E-02	4.57E+01	6.41E+00	3.55E-01	1.22E+01
			BA-SAM	1.36E-01	5.87E+00	**1.03E+00**	5.63E-01	1.24E+00
f_2	Zakharov	10	BA	3.51E+01	3.13E+02	1.30E+02	9.41E+01	8.31E+01
			NABA	1.54E-03	1.27E+05	4.22E+03	1.81E-01	2.31E+04

			BA-SAM	1.49E-03	3.87E+00	**3.02E-01**	1.41E-01	6.94E-01
		30	BA	4.70E+02	1.30E+09	4.32E+07	9.62E+02	2.37E+08
			NABA	5.27E+00	1.30E+09	4.32E+07	1.78E+01	2.37E+08
			BA-SAM	1.08E+01	6.90E+02	**1.05E+02**	5.53E+01	1.37E+02
		50	BA	1.22E+03	2.65E+03	2.19E+03	2.24E+03	3.60E+02
			NABA	5.29E+01	1.95E+03	**5.20E+02**	3.74E+02	4.96E+02
			BA-SAM	1.98E+02	1.54E+03	5.83E+02	4.76E+02	3.85E+02
f$_3$	Step	10	BA	6.98E+02	4.73E+03	2.19E+03	2.21E+03	1.01E+03
			NABA	0.00E+00	3.60E+03	8.99E+02	5.43E+02	1.04E+03
			BA-SAM	0.00E+00	1.75E+02	**8.23E+00**	0.00E+00	3.37E+01
		30	BA	4.65E+03	1.71E+04	1.00E+04	9.72E+03	3.00E+03
			NABA	2.19E+02	1.26E+04	5.07E+03	5.44E+03	3.32E+03
			BA-SAM	1.00E+01	5.33E+03	**6.57E+02**	1.16E+02	1.27E+03
		50	BA	1.22E+04	2.64E+04	1.82E+04	1.78E+04	3.89E+03
			NABA	2.12E+03	2.38E+04	1.02E+04	9.32E+03	5.43E+03
			BA-SAM	1.91E+02	8.17E+03	**1.59E+03**	6.68E+02	1.98E+03
f$_4$	Quartic	10	BA	2.14E+00	4.91E+00	3.45E+00	3.31E+00	7.35E-01
			NABA	7.11E-01	4.57E+00	3.11E+00	3.40E+00	1.03E+00
			BA-SAM	1.66E+00	4.39E+00	**2.85E+00**	2.82E+00	8.31E-01
		30	BA	1.04E+01	2.64E+01	1.65E+01	1.63E+01	2.80E+00
			NABA	9.35E+00	2.09E+01	1.48E+01	1.48E+01	2.85E+00
			BA-SAM	9.90E+00	2.12E+01	**1.45E+01**	1.47E+01	3.29E+00
		50	BA	2.47E+01	4.42E+01	3.29E+01	3.22E+01	4.94E+00
			NABA	1.86E+01	4.71E+01	3.18E+01	3.22E+01	7.02E+00
			BA-SAM	1.80E+01	5.12E+01	**2.85E+01**	3.04E+01	7.15E+00
f$_5$	Schwefel	10	BA	-3.05E+03	-1.04E+03	-1.84E+03	-1.68E+03	5.05E+02
			NABA	-2.74E+03	-1.37E+03	-1.99E+03	-1.94E+03	3.46E+02
			BA-SAM	-3.61E+03	-1.82E+03	**-2.81E+03**	-2.82E+03	4.46E+02
		30	BA	-8.21E+03	-2.08E+03	-4.29E+03	-3.59E+03	1.83E+03
			NABA	-5.65E+03	-2.48E+03	-3.72E+03	-3.62E+03	7.99E+02
			BA-SAM	-8.64E+03	-4.15E+03	**-6.81E+03**	-6.97E+03	1.11E+03
		50	BA	-1.37E+04	-2.74E+03	-7.21E+03	-7.73E+03	2.84E+03
			NABA	-6.69E+03	-3.57E+03	-4.83E+03	-4.73E+03	7.55E+02
			BA-SAM	-1.32E+04	-5.95E+03	**-1.03E+04**	-1.03E+04	1.75E+03
f$_6$	Rosenbrock	10	BA	1.50E+03	1.22E+05	4.10E+04	2.60E+04	3.66E+04
			NABA	5.87E-01	4.89E+04	2.55E+03	1.31E+01	1.00E+04
			BA-SAM	2.72E+00	4.70E+02	**3.24E+01**	1.03E+01	8.75E+01
		30	BA	7.38E+04	1.38E+06	3.78E+05	3.47E+05	2.56E+05
			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
		50	BA	1.96E+05	2.06E+06	8.14E+05	7.14E+05	4.48E+05
			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
f$_7$	Rastrigin	10	BA	8.59E+01	1.94E+02	1.27E+02	1.28E+02	2.53E+01
			NABA	1.34E+01	5.08E+01	2.85E+01	2.68E+01	9.33E+00
			BA-SAM	6.73E+00	3.83E+01	**2.33E+01**	2.56E+01	8.85E+00
		30	BA	4.04E+02	6.07E+02	4.87E+02	4.81E+02	5.60E+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
		50	BA	7.05E+02	1.13E+03	8.89E+02	8.71E+02	9.97E+01
			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
f$_8$	Griewangk	10	BA	6.81E+00	4.15E+01	1.86E+01	1.46E+01	9.19E+00
			NABA	2.68E+00	5.06E+01	1.74E+01	1.41E+01	1.21E+01
			BA-SAM	1.31E-01	1.11E+01	**2.72E+00**	1.17E+00	2.98E+00
		30	BA	4.61E+01	1.75E+02	8.89E+01	8.77E+01	2.46E+01
			NABA	4.67E+01	1.70E+02	8.56E+01	8.21E+01	2.63E+01
			BA-SAM	7.18E+00	8.68E+01	**3.38E+01**	2.67E+01	2.17E+01
		50	BA	7.35E+01	2.59E+02	1.61E+02	1.63E+02	4.18E+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
f$_9$	Ackley	10	BA	1.10E+01	1.68E+01	1.34E+01	1.35E+01	1.40E+00
			NABA	1.20E-01	1.15E+01	3.10E+00	4.96E-01	4.36E+00
			BA-SAM	2.09E-02	6.44E+00	**9.93E-01**	2.68E-01	1.54E+00

		30	BA	1.29E+01	1.76E+01	1.55E+01	1.55E+01	1.03E+00
			NABA	3.19E-01	1.42E+01	6.51E+00	4.39E+00	5.14E+00
			BA-SAM	1.76E+00	1.21E+01	**4.09E+00**	3.59E+00	2.45E+00
		50	BA	1.35E+01	1.77E+01	1.57E+01	1.55E+01	9.41E-01
			NABA	3.21E+00	1.55E+01	1.03E+01	1.18E+01	4.01E+00
			BA-SAM	3.85E+00	1.08E+01	**6.68E+00**	6.28E+00	1.91E+00
f_{10}	Penalized	10	BA	2.56E+01	1.11E+06	8.90E+04	3.75E+03	2.16E+05
			NABA	1.44E-03	5.29E+03	2.68E+02	6.16E+00	1.06E+03
			BA-SAM	8.28E-05	5.47E+00	**3.16E-01**	1.25E-03	1.04E+00
		30	BA	2.91E+04	9.35E+06	1.83E+06	1.14E+06	2.10E+06
			NABA	1.40E+00	9.21E+05	6.09E+04	1.15E+01	1.95E+05
			BA-SAM	2.25E+00	6.82E+05	**2.28E+04**	5.61E+00	1.25E+05
		50	BA	8.47E+04	2.87E+07	5.59E+06	2.94E+06	6.06E+06
			NABA	3.40E+00	3.42E+06	3.52E+05	4.51E+01	8.64E+05
			BA-SAM	5.32E+00	9.43E+03	**3.97E+02**	1.50E+01	1.73E+03

The results in Table 2 clearly demonstrate that the proposed algorithm — BA-SAM outperforms both BA and NABA on all benchmark test functions for all the dimensionalities. However, the performance of all three algorithms seems to drop with the increasing dimensions. This is quite rational, because the size of the search space increases exponentially with the increasing dimensionality *d*. Among the three algorithms, the performance of the original Bat algorithm (BA) is quite poor. The reason is —in the original Bat Algorithm, the loudness *A* decreases quickly with the exponential increment of the pulse emission rate *r* [**13**]. This causes the algorithm to drastically lose the degree of explorations during the late generations (because, the condition at the 8th line of the algorithm pseudo-code is very less likely to be satisfied as the generations proceed). For this reason, the exploration capability of BA drastically decreases and the algorithm is likely to be trapped around the locally optimal points of the search space. Also, it can be observed in Table 2 that the accuracy of the results is relatively higher for the unimodal functions compared to the multimodal functions [**13**]. This is due to the reason that the multimodal functions have many locally optimal points and only a single global optimum. This makes their optimization very challenging, because escaping from the local optima and reaching the neighborhood of the single global optimum may be extremely difficult. Fig. 5 presents the convergence graphs for all three algorithms for a unimodal function (i.e., sphere function f_1) and a multimodal function (Ackley function f_9). For both the functions, BA-SAM shows higher convergence speed than both NABA and BA and reaches very close proximity of the global minimum ($F_{\min} = 0$). In contrast, NABA gets trapped around several locally minimal points before it reaches close to the global minimum, while BA gets trapped and soon prematurely converges around some poor local minima, far from the global minimum. Thus the convergence characteristics of BA-SAM is much better than both of its competitors — BA and NABA.

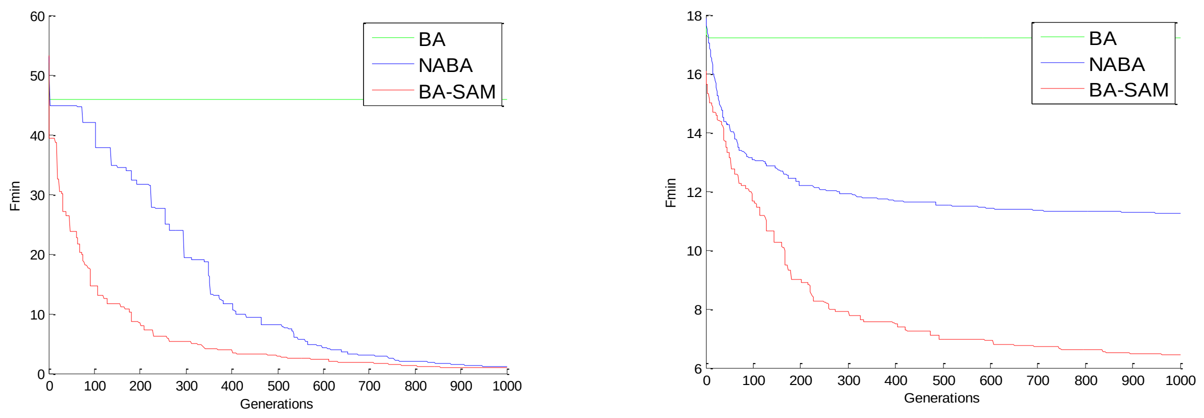

Fig. 5: Convergence characteristics of BA, NABA and BA-SAM on the Sphere function f_1 and Ackley function f_9

5. CONCLUSION AND FUTURE WORKS

In this paper, we have introduced a novel swarm intelligence based algorithm — BA-SAM and evaluated it using a number of benchmark problems on numeric optimization. BA-SAM employs a self-adaptive scheme to determine the best mutation strategy by using both Gaussian and Cauchy distributions for mutations. BA-SAM has been implemented and tested on several state-of-the-art benchmark optimization problems. The results (i.e., both final solution quality and convergence characteristics) clearly demonstrate that BA-SAM is better or at least comparable to the original Bat algorithm and another improved variant of the Bat algorithm. Hence, the BA-SAM may be a good alternative to deal with complex numerical optimization problems. Further research can compare the performance of other existing meta-heuristic and swarm intelligence based algorithms with BA-SAM and can observe their convergence characteristics to discover their relative strengths and weaknesses across different and diverse search spaces. There is also research opportunity to determine the best settings of the control parameters of the Bat-based algorithms, such as the pulse rate *r* and loudness *A*. Also, we have not sufficiently experimented with different population sizes and other parameter settings of BA-SAM, nor did we make the population size and the other parameters adaptive. We plan to include these experiments in our future research with BA-SAM.

6. REFERENCES

[1] Hsiang-Cheh Huang, John F. Roddick, Jeng-Shyang Pan Shu-Chuan Chu, "Overview of Algorithms for Swarm Intelligence ," in *ICCCI 2011, Part I, LNCS 6922, pp. 28–41*, Kaohsiung, 2011.

[2] J Kennedy and R Eberhart, "Practicle swarm optimization," in *IEEE International Conference Neural Networks*, Perth, Australia, 1995, pp. 1942-1945.

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

[4] 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.

[5] John D Altringham, *Bats: Biology and Behaviour*.: Oxford University Press, 1996.

[6] Md. Wasi Ul Kabir, Md. Monirul Islam Mohammad Shafiul Alam, "On the Performance of Recurring Multistage Evolutionary Algorithm for Continuous Function Optimization," in *International Conference onComputer and Information Technology* , Dhaka, 2010.

[7] Yong Liu, Guangming Lin Xin Yao, "Evolutionary Programming Made Faster," in *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 3, NO. 2*, 1999.

[8] 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.

[9] P Richardson, "Bats," London, 2008.

[10] Amr Rekaby, "Directed Artificial Bat Algorithm (DABA) A New Bio-Inspired Algorithm," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Cairo, 2013.

[11] Banu, 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.

[12] E. Ugur Kucuksille, Y. Cengiz S. Yılmaz, "Modified Bat Algorithm," *ELEKTRONIKA IR ELEKTROTECHNIKA*, vol. 20, no. ISSN 1392-1215, p. 2, 2014.

[13] Nazmus Sakib, Syed Mustafizur, Mohammad Shafiul Alam Md. Wasi Ul Kabir, "A Novel Adaptive Bat Algorithm to Control Explorations nd Exploitations for Continuous Optimization Problems," *International Journal of Computer Applications* , vol. 94, no. 13, 2014.

[14] L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa R. Y. M. Nakamura, "BBA: A Binary Bat Algorithm for Feature Selection," in *25th Conference on Graphics, Patterns and Images*, 2012.

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

[16] S, Yang X, "Nature-inspired Metaheuristic Algorithms," 2008.

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

[18] Koffka Khan and Ashok 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.