Modified Bat Algorithm With Cauchy Mutation and Elite Opposition-Based Learning

Fábio A. P. Paiva
Academic Direction
Federal Institute of Rio Grande do Norte
Parnamirim/Brazil
fabio.procopio@ifrn.edu.br

Cláudio R. M. Silva
Department of Communications Engineering
Federal University of Rio Grande do Norte
Natal/Brazil
claudio.rmsilva@gmail.com

Izabele V. O. Leite
Department of Chemical Engineering
Federal University of Rio Grande do Norte
Natal/Brazil
izaleite11@hotmail.com

Marcos H. F. Marcone
School of Science and Technology
Federal University of Rio Grande do Norte
Natal/Brazil
marcosmarcone48@gmail.com

***Abstract*—Metaheuristics can be used to solve optimization complex problems because they offer approximate and acceptable solutions. In recent years, nature has been a source of inspiration for many computer scientists when proposing new metaheuristics such as the algorithms inspired by swarm intelligence. They are based on the behavior of animals that live in groups such as birds, fishes and bats. In this context, Bat Algorithm (BA) is a recent metaheuristic inspired by echolocation of bats during their flights. However, a problem that this algorithm faces is the loss of the ability to generate diversity and, consequently, the chances of finding the global solution are reduced. This paper proposes a modification to the original BA using two methods known as Cauchy mutation operator and Elite Opposition-Based Learning. The new variant aims to preserve the diversity of the algorithm and increases its convergence velocity. It was compared to the original BA and another variant found in the literature. For this comparison, the proposed variant used four benchmark functions, during 30 independent runs. After the experiments, the superiority of the new variant is highlighted when the results are compared to the original BA.**

***Index Terms*—Metaheuristic, Swarm Intelligence, Bat Algorithm, Elite Opposition-Based Learning, Cauchy Mutation**

I. Introduction

Optimization problems are present in many real-world domains such as localization problem [1], parallel processing [2], multiprocessor task scheduling [3], air traffic management [4] and many others. Optimization is a process that aims to find the best solution for a given problem. Therefore, there is a constant need to develop efficient and robust algorithms capable of solving real problems. Metaheuristic optimization algorithms have been widely used to solve these types of problems.

Metaheuristics can be used to guide a subordinate heuristic by combining different concepts for exploring and exploiting the search space. Some metaheuristics are inspired by nature, for example, swarm intelligence. Therefore, they are known as algorithms based on swarm intelligence.

In recent years, various metaheuristic algorithms based on swarm intelligence have been proposed. For instance, Bat Algorithm (BA) [5] is inspired by the echolocation of bats using a frequency adjustment technique to balance exploration and exploitation. BA has been used to resolve many real-world problems such as image processing [6], data classification [7] and distribution systems [8].

A problem commonly faced by metaheuristic algorithms is known as premature convergence and it occurs when the algorithm stays "trapped" in local optimum. BA is powerful in local searches, but at times it be may trap into some local optima, so that it cannot perform global search well.

Various papers have been proposed to improve the original BA performance. Hybrid Bat Algorithm (HBA) [9] was obtained by hybridizing the original BA with Differential Evolution [10] strategies. In [11], a variant called Bat Algorithm with Self-Adaptive Mutation (BA–SAM) was proposed. Differently from the original BA which generates new solutions from random walks only, BA–SAM produces the mutation step size using both Gaussian and Cauchy distributions. In [12], an Improved BA (IBA) is presented proposing three modifications: Inertia Weigh Factor Modification, Adaptive Frequency Modification and Scout Bee Modification.

This paper proposes a BA variant that uses two approaches called Cauchy mutation and elite opposition-based learning to increase the diversity of the population. They can be used to improve the convergence velocity of the original BA. The novel variant and original BA are compared by means of four benchmark functions. The results show that the new variant has achieved better performance compared to the original BA.

The paper is organized as follows. Section II presents a brief background about BA and elite opposition-based learning. In Section III, a novel variant is proposed to improve performance of the original BA. In Section IV, the experiments are presented and the results obtained are analyzed. Finally, Section V presents conclusions and future work.

II. Background

This section presents some needed information to understand the algorithm proposed in Section III, such as Bat Algorithm and the concept of Elite Opposition-Based Learning.

A. Bat Algorithm

Bio-inspired metaheuristic algorithms have been receiving much attention. Yang [5] proposed one of these algorithms which is inspired by echolocation and it is known as Bat Algorithm (BA). Echolocation is a sophisticated biological capability that the bats use to detect their preys and to avoid obstacles.

Bats make calls as they fly and listen to the returning echoes to build up a sonic map of their surroundings. Thus, a bat can compute how far they are from an object. Furthermore, bats can distinguish the difference between an obstacle and a prey even in complete darkness [13].

In BA, all bats use echolocation to perceive and calculate distances. The ith bat has a position x_i (solution), velocity v_i and frequency f_i. For each iteration t, the ith bat moves toward the current best position. In order to implement global search, the position, the velocity, and the frequency of the ith bat are updated, according to Equation (1):

$$
\begin{aligned}
f_i &= f_{min} + (f_{max} - f_{min}) \cdot \beta \\
v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_g^{t-1}) \cdot f_i \\
x_i^t &= x_i^{t-1} + v_i^t,
\end{aligned}
\tag{1}
$$

where f_{min} and f_{max} are minimum and maximum frequencies, respectively. β is a random value generated from a uniform distribution $\in$ [0, 1] and x_g^{t-1} represents the current global best solution.

In order to implement local search, a new candidate solution is generated apart from or close to the current best solution as follows:

$$x_{new} = x_g^{t-1} + \epsilon \bar{A}^t, \tag{2}$$

where ϵ is a random value $\in$ [-1, 1] and $\bar{A}^t$ is the average value of the loudness of all bats.

According to the proximity of the target, the bats decrease the loudness $A_i \in [A_0, A_{min}]$ and increase the rate of pulse emission $r_i \in [0, 1]$ as follows:

$$
\begin{aligned}
A_i^{t+1} &= \alpha A_i^t \\
r_i^{t+1} &= r_i^0[1 - exp(-\gamma t)],
\end{aligned}
\tag{3}
$$

where α and γ are constants. For any $0 < \alpha < 1$ and $\gamma > 0$, then

$$
\begin{aligned}
&A_i^t \to 0 \\
&r_i^t \to r_i^0, \quad as\ t \to \infty.
\end{aligned}
\tag{4}
$$

The pseudocode of BA can be summarized as shown in Algorithm 1. It is initialized with a random generation of the population (line 1). In line 2, the initial frequency f_i is determined for the position X_i. Line 3 initializes the pulse rate r_i and the loudness A_i. Lines 4–16 represent the evolution of bats over time. In this evolution, new solutions are established and the frequencies, velocities and positions are updated (line 6). In the next line, the pulse emission rate is compared to a random real number drawn from a uniform distribution $\in [0, 1]$. When the random number is greather than r_i, a new solution is generated around the best solution (line 9). In line 11, it is evaluated if this solution will be accepted. If positive, r_i is increased and A_i is decreased (line 12). Finally, the bats are ranked and the best one is selected (line 14).

Algorithm 1 BA Pseudocode

1: Initialize the bats population X_i $i = (1, 2, ..., N)$ and V_i
2: Define pulse frequency f_i at X_i
3: Initialize pulse rates r_i and the loudness A_i
4: **while** $(t \leq num\_max\_iter)$ **do**
5:   **for** $i = 1 : X$ **do**
6:     Generate new solutions by adjusting frequency, and updating velocities and locations/solutions
7:     **if** $(rand > r_i)$ **then**
8:       Select the best solution
9:       Generate a local solution around the best solution
10:     **end if**
11:     **if** $(rand < A_i\ \&\ fit(x_i) < fit(x_g^{t-1}))$ **then**
12:       Increase r_i and reduce A_i
13:     **end if**
14:     Rank the bats and find the current best
15:   **end for**
16: **end while**

B. Elite Opposition-Based Learning

In general, random solutions are used to initialize metaheuristic algorithms. Over time, the algorithm evolves towards global optimum and it is common that the solutions are distant from the one that is considered optimal. The worst case occurs when the solutions are in the opposite position to the optimal one. An alternative is to search simultaneously in all directions or to search in the opposite direction. Elite Opposition-Based Learning (EOBL) [14] can be applied to deal with this situation.

Before introducing EOBL, the concept of Opposition-Based Learning (OBL) [15] will be explained. In the computational intelligence field, OBL have been used to increase the algorithms efficiency. For a given problem, OBL evaluates simultaneously a solution x and its respective opposite solution $\tilde{x}$, which allows a candidate solution to be found close to the global optimum.

OBL can be apllied not only to generate random initial solutions, but also at each iteration to improve the diversity during the exploration process [16]. Next, the definitions for Opposite Number and Opposite Point are presented.

***Definition 1*: Opposite Number** – Let $x \in \Re$, $x \in [a, b]$, the opposite number $\tilde{x}$ is defined by Equation (5):

$$\tilde{x} = a + b - x. \tag{5}$$

Similarly, the same definition can be extended for multidimensional problems.

***Definition 2*: Opposite Point** – Let $P = (x_1, x_2, ..., x_n)$ be a point in the n-dimensional space, where x_1, x_2, ..., $x_n \in \Re$ and $x_i \in [a_i, b_i]$, $\forall i \in \{1, 2, ..., n\}$. Therefore, the opposite point $\tilde{P} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ is defined by its components:

$$\tilde{x_i} = a_i + b_i - x_i. \tag{6}$$

Considering an optimization problem, P is given as a point in the n-dimensional space (i.e., a candidate solution) and $f(\cdot)$ is a function used to measure the fitness value of the candidate solution. According to Definition 2, $\tilde{P} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ is the opposite point of $P = (x_1, x_2, ..., x_n)$. Therefore, in an optimization problem, if $f(\tilde{P}) \leq f(P)$ to minimize $f(\cdot)$, then P will be replaced by $\tilde{P}$; otherwise, solution P remains.

Once the OBL definition has been presented, an example will be used to explain EOBL. In this paper, the bat with the best fitness value is viewed as the elite individual. Supposing that the elite individual is $X_e = (x_{e1}, x_{e2}, ...x_{en})$, the elite opposition-based solution for the individual $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ can be defined as $\hat{X}_i = (\hat{x}_{i1}, \hat{x}_{i2}, ...\hat{x}_{in})$. It can be obtained by Equation (7):

$$\hat{x}_{ij} = \delta \cdot (da_j + db_j) - x_{ej}, i = 1, 2, ..., N; j = 1, 2, ..., n \tag{7}$$

where N is the population size, n is the dimension of X, $\delta \in (0, 1)$, (da_j, db_j) is the dynamic bound of jth decision variable. The dynamic bound can be obtained by the following equation:

$$da_j = min(x_{ij}), \quad db_j = max(x_{ij}) \tag{8}$$

Dynamic bound can make $\hat{x}_{ij}$ jump out of (da_j, db_j). If it happens, Equation (9) will be used to reset $\hat{x}_{ij}$:

$$\hat{x}_{ij} = rand(da_j, db_j), \quad \hat{x}_{ij} < da_j \mid\mid \hat{x}_{ij} > db_j. \tag{9}$$

III. Modified BA with Cauchy Mutation and EOBL

This paper proposes a novel BA variant that uses two techniques known as Cauchy mutation and EOBL. They can be used to increase the exploration of the search space and consequently diversify the population. The proposed variant is an alternative for the original BA to achieve better performance and higher velocity of convergence.

Before presenting the new variant, it is important to highlight the Cauchy mutation. It is characterized by having a density function of one-dimensional probability centered on the origin that is defined by:

$$f_T(x) = \frac{1}{\pi} \frac{T}{(T^2 + x^2)} \quad -\infty < x < \infty, \tag{10}$$

where $T > 0$ is a scalar parameter. The corresponding distribution function is given by:

$$F_T(x) = \frac{1}{2} + \frac{1}{\pi} \arctan(\frac{x}{T}). \tag{11}$$

Therefore, it can be observed that the variation of the Cauchy distribution is infinite. In thesis, this allows a global search with more efficiency [17] and, as of a consequence, better results.

The proposed variant is presented in Algorithm 2. It is verified that the main differences related to the original BA are in lines 14–20. In line 14, a bat is randomly drawn and it is called bat_r. Next, EOBL is applied on bat_r and it is verified if its fitness value is better than the original value. If this condition is true, the changes on bat_r will be commited (line 16).

In line 18, Cauchy mutation is applied on bat_r and after mutation, it is verified if its new fitness value is better than the current fitness value. If this condition is true, bat_r will receive the changes perfomed through Cauchy mutation (line 19). The rest of the algorithm is similar to the original BA.

Algorithm 2 Modified BA Pseudocode

1: Initialize the bat population X_i $i = (1, 2, ..., N)$ and V_i
2: Define pulse frequency f_i at X_i
3: Initialize pulse rates r_i and the loudness A_i
4: **while** $(t \leq num\_max\_iter)$ **do**
5:   **for** $i = 1 : X$ **do**
6:     Generate new solutions by adjusting frequency, and updating velocities and locations/solutions
7:     **if** $(rand > r_i)$ **then**
8:       Select the best solution
9:       Generate a local solution around the best solution
10:     **end if**
11:     **if** $(rand < A_i \ \& \ fit(x_i) < fit(x_g^{t-1}))$ **then**
12:       Increase r_i and reduce A_i
13:     **end if**
14:     $bat_r \leftarrow random\_draw()$
15:     **if** $fit(EOBL(bat_r)) \leq fit(bat_r)$ **then**
16:       $bat_r \leftarrow EOBL(bat_r)$
17:     **end if**
18:     **if** $fit(cauchy\_mutation(bat_r)) < fit(bat_r)$ **then**
19:       $bat_r \leftarrow cauchy\_mutation(bat_r)$
20:     **end if**
21:     Rank the bats and find the current best
22:   **end for**
23: **end while**

IV. Computational Experiments

This section presents benchmark functions used to validade the proposed algorithm. It also presents the configuration of the experiments as well as the values defined to the algorithm parameters. Finally, several simulations are presented and the results analisys is performed.

A. Benchmark Functions and Experimental Setup

It is common to use benchmark functions with the assumption that their difficulty corresponds to the difficulties encountered in real-world applications. Therefore, they are used to validate and compare the optimization algorithms, as well as to validate any global optimization approach [18].

Four benchmark functions were chosen to perform the experiments. They are often applied in minimization problems and have been used in various BA studies [9][11][19]. For each one of them, the formulation, the search space, the optimal solution and the modality are presented in Table I.

Sphere and Rosenbrock functions are unimodal. Often they are used to test the ability of the algorithm in local searches. On the other hand, Griewank and Rastrigin have several local minima and they are used to verify the ability of the algorithm in global searches.

All routines were implemented in the MATLAB programming language. The experiments were run on a computer that uses Intel Core i7 processor with 2.4 GHz frequency, 8 GB of RAM and Windows 10 Home Single Language (64 bits). Multiprocessing techniques were not used.

The experiments use various configurations to compare the original BA to the proposed variant. Dimensions of the problem and the number of iterations are varied. For each run, the number of iterations is set 300, 600 and 900 for 10 dimensions, 20 dimensions and 30 dimensions respectively. The variant is also compared to one of the literature using 30 dimensions and 900 iterations. The algorithms are tested with 30 independent runs and the number of bats is fixed to 50.

The values defined for the parameters used in the experiments were the same used by [20]: loudness A_i and pulse rate r_i are fixed to 0.5, initial loudness A_0 equals 2 and minimum loudness A_{min} equals 0. The values defined for the constants γ and α are 0.05 and 0.95, respectively.

B. Simulations and Results

Figure 1 shows the average convergence process to optimizate Sphere function. It is observed that original BA and modified BA stagnate near iteration 90 and 650, aproximately. However, the proposed variant has better results than BA.

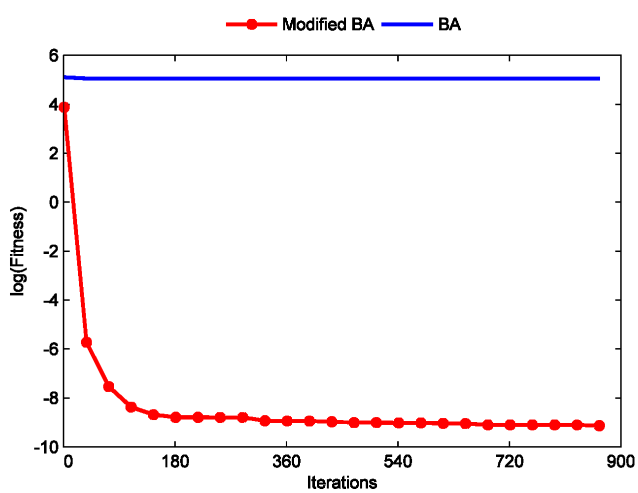

Figure 1. Convergence mean behavior for f_1: 50 bats, 30D, 900 iterations.

In Figure 2, BA stagnates at the beginning of iterations and the proposed variant presents a slow convergence velocity. It is observed that the convergence velocity of the modified BA begins next to iteration 40 and remains until the end.

Figure 3 represents the optimization of the Griewank function. Again, BA presents an insufficient performance and a stagnant behavior at the beginning of iterations. However, the proposed algorithm presents an active convergence behavior. In some runs, according to Table II, the variant found the optimal solution. That testifies its superiority over the BA.

In Figure 4, original BA has stagnated for a large number of iterations. On the other hand, the proposed variant presents an active behavior until near iteration 620. As in the optimization of the Griewank function, in some runs, the proposed variant found the optimal solution according to Table II.

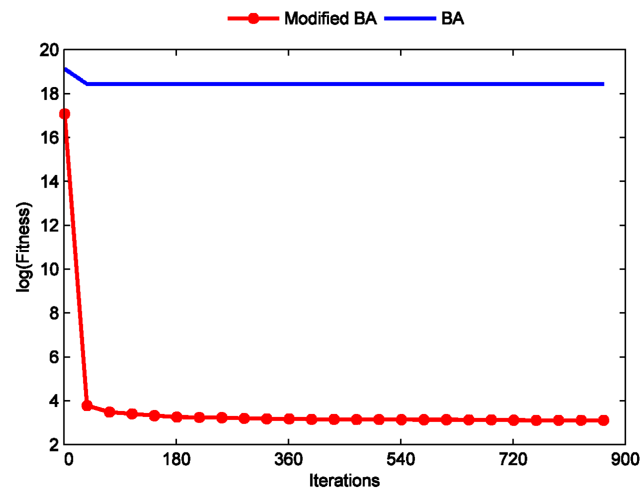

Figure 2. Convergence mean behavior for f_2: 50 bats, 30D, 900 iterations.

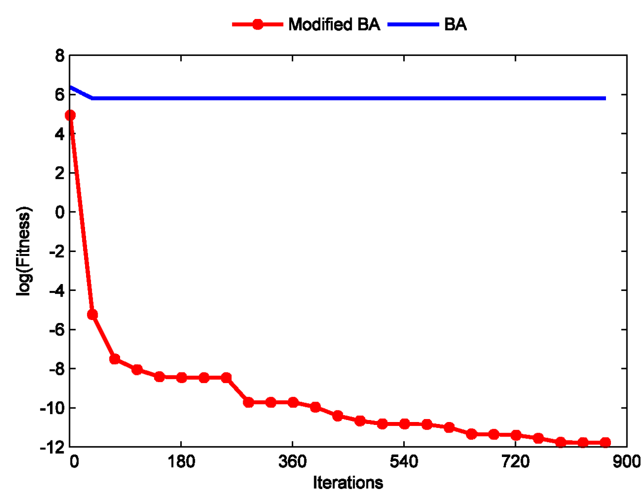

Figure 3. Convergence mean behavior for f_3: 50 bats, 30D, 900 iterations.

Wilcoxon Test [21] was applied with significance level 0.05 to evaluate the solutions found by original BA and the proposed variant. Wilcoxon is a non-parametric statistical test used to compare two independent samples. The null hypothesis H_0 indicates that two samples come from the same population, whereas the alternative hypothesis H_1 indicates that one has higher values than the other. When *p-value* is less than the

Table I
EVALUATED BENCHMARK FUNCTIONS

Function	Formulation	Search Space	Optimal Solution	Modality
Sphere	$f_1(x) = \sum_{i=1}^{d} x_i^2$	$-100 \leq x_i \leq 100$	$f(x^*) = 0$	Unimodal
Rosenbrock	$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$	$f(x^*) = 0$	Unimodal
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \leq x_i \leq 600$	$f(x^*) = 0$	Multimodal
Rastrigin	$f_4(x) = \sum_{i=1}^{d} [x_i^2 - 10cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	$f(x^*) = 0$	Multimodal

Table II
COMPARISON BETWEEN ORIGINAL BA AND MODIFIED BA WITH CAUCHY MUTATION AND EOBL (BEST RESULTS IN BOLD).

Function	Dimension	Iteration	Original BA			Modified BA with Cauchy and EOBL			p-Value
			Best	Worst	Average (Std. Dev.)	Best	Worst	Average (Std. Dev.)	
f_1	10	300	8.6172e+00	4.8308e+01	3.1324e+01 (9.6723e+00)	1.1594e-09	3.0523e-03	**3.6649e-04 (7.2575e-04)**	1.8626e-09
	20	600	3.8845e+01	1.2442e+02	9.7476e+01 (1.6938e+01)	4.6546e-20	4.6571e-03	**2.4059e-04 (9.116223e-04)**	1.8626e-09
	30	900	1.0627e+02	1.8766e+02	1.5597e+02 (2.1357e+01)	3.5104e-40	3.1818e-03	**1.0790e-04 (5.8060e-04)**	1.8626e-09
f_2	10	300	2.4634e+06	4.2601e+07	1.2908e+07 (1.0442e+07)	5.4910e-02	9.3032e+00	**5.5290e+00 (3.8021e+00)**	1.8626e-09
	20	600	1.7592e+07	1.4977e+08	7.1782e+07 (2.8333e+07)	6.0796e-02	4.4866e+01	**1.5756e+01 (1.0521e+01)**	1.8626e-09
	30	900	4.4690e+07	2.6762e+08	9.9976e+07 (4.5869e+07)	3.9020e-03	3.5483e+01	**2.2355e+01 (1.1963e+01)**	1.8626e-09
f_3	10	300	2.3941e+01	1.3502e+02	6.7785e+01 (2.7158e+01)	2.7503e-10	1.2630e-02	**8.6462e-04 (2.3687e-03)**	1.8626e-09
	20	600	1.1565e+02	3.3244e+02	2.0068e+02 (4.8167e+01)	0	3.4703e-03	**1.6720e-04 (6.4169e-04)**	1.8626e-09
	30	900	1.3774e+02	5.2320e+02	3.2714e+02 (8.9976e+01)	0	1.1120e-04	**7.6489e-06 (2.3962e-05)**	1.8626e-09
f_4	10	300	5.2910e+01	1.2829e+02	9.3940e+01 (1.8430e+01)	4.3541e-05	3.2765e+00	**2.3527e-01 (6.4058e-01)**	1.8626e-09
	20	600	1.9263e+02	2.9572e+02	2.5651e+02 (2.1655e+01)	0	7.2776e+01	**9.2916e+00 (2.1694e+01)**	1.8626e-09
	30	900	3.2302e+02	4.5690e+02	4.0736e+02 (2.7965e+01)	0	1.3339e+02	**2.9885e+01 (5.0245e+01)**	1.8626e-09

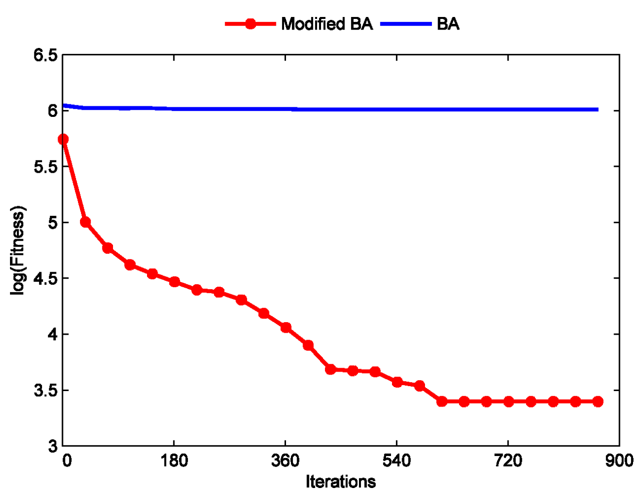

Figure 4. Convergence mean behavior for f_4: 50 bats, 30D, 900 iterations.

significance level then it decides to reject H_0, i.e., there is significant difference between the samples.

Table II compares the algorithms performance numerically. For each evaluated function, the table shows problem dimension, number of iterations, best and worst solutions, average and standard deviation, and p-value from Wilcoxon Test.

The modified BA is also compared to a variant of the literature named IBA [12]. In this comparison, in order that no algorithm will benefit from a suitable configuration, the same configuration defined in [20] is used: 50 bats, 30 dimensions and 900 iterations, during 30 independent runs. After the experiments, the average rate of the fitness values and the standard deviation are shown in Table III.

The results show that the modified BA exceeds both the original BA and the IBA variant, as shown in Tables II and III. When the modified BA is compared to the original BA and IBA, the results show that the proposed variant presents stability, better solutions and higher convergence velocity in the optimization process.

Table III
PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS.

Function	Original BA	IBA	Modified BA
f_1 – Sphere	1.5597e+02 (2.1357e+01)	1.3493e-03 (1.6313e-04))	**1.0790e-04 (5.8060e-04)**
f_2 – Rosenbrock	9.9976e+07 (4.5869e+07)	1.0208e+02 (1.4067e+02)	**2.2355e+01 (1.1963e+01)**
f_3 – Griewank	3.2714e+02 (8.9976e+01)	6.3576e+01 (1.6637e+01)	**7.6489e-06 (2.3962e-05)**
f_4 – Rastrigin	4.0736e+02 (2.7965e+01)	1.4073e+02 (3.8512e+01)	**2.9885e+01 (5.0245e+01)**

V. CONCLUSION

This paper presented a novel variant for the metaheuristic that is based on bats echolocation process. The variant consists of combining original BA with Cauchy mutation and elite opposition-based learning that were used to improve the convergence velocity of the original algorithm. The modified BA generated diversity of solutions and thus better results were achieved.

The modified BA was compared to the original BA and a variant called IBA. Thus, several computational experiments were performed using four classic benchmark functions: Sphere, Rosenbrock, Griewank and Rastrigin. The presented results showed the superiority of the proposed variant related to the original BA and the IBA variant. They also showed that the combination Cauchy mutation and elite opposition-based learning preserved the diversity of the original algorithm.

As future work, it is intended to evaluate the performance of the modified BA in real applications of chemical engineering. In addition, it is also intended to implement a new approach for local search to further balance exploration and exploitation of the algorithm.

REFERENCES

[1] A. O. de Sá, N. Nedjah, and L. de Macedo Mourelle, “Genetic and backtracking search optimization algorithms applied to localization problems,” in *International Conference on Computational Science and Its Applications*, 2014, pp. 738–746.

[2] A. J. C. C. Lins, F. B. Lima-Neto, and C. J. A. Bastos-Filho, “Paralelização de algoritmos de busca baseados em cardumes através de unidades de processamento gráfico,” *Learning and Nonlinear Models*, vol. 13, 2015.

[3] J. A. Núñez-Humán and Y. Túpac, “Multiprocessor task scheduling on heterogeneous environments by a hybrid chemical reactions optimization,” in *IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2016, pp. 1–6.

[4] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín-Campo, “Multiobjective optimization for aircraft conflict resolution. a metaheuristic approach,” *European Journal of Operational Research*, vol. 248, no. 2, pp. 691–702, 2016.

[5] X.-S. Yang, “A new metaheuristic bat-inspired algorithm,” in *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[6] J. W. Zhang and G. G. Wang, “Image matching using a bat algorithm with mutation,” in *Applied Mechanics and Materials*, vol. 203, 2012, pp. 88–93.

[7] S. Mishra, K. Shaw, and D. Mishra, “A new meta-heuristic bat inspired classification approach for microarray data,” *Procedia Technology*, vol. 4, pp. 802–806, 2012.

[8] V. U. Reddy and A. Manoj, “Optimal capacitor placement for loss reduction in distribution systems using bat algorithm,” *IOSR journal of Engineering*, vol. 2, no. 10, pp. 23–27, 2012.

[9] I. Fister Jr, D. Fister, and X.-S. Yang, “A hybrid bat algorithm,” *arXiv preprint arXiv:1303.6310*, 2013.

[10] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[11] M. W. U. Kabir and M. S. Alam, “Bat algorithm with self-adaptive mutation: a comparative study on numerical optimization problems,” *International Journal of Computer Applications*, vol. 100, no. 10, 2014.

[12] S. Yilmaz and E. U. Kucuksille, “Improved bat algorithm (iba) on continuous optimization problems,” *Lecture Notes on Software Engineering*, vol. 1, no. 3, p. 279, 2013.

[13] R. Y. Nakamura, L. A. Pereira, K. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, “Bba: a binary bat algorithm for feature selection,” in *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2012, pp. 291–297.

[14] X. Zhou, Z. Wu, and H. Wang, “Elite opposition-based differential evolution for solving large-scale optimization problems and its implementation on gpu,” in *13th Int. Conf. on Parallel and Distributed Computing, Applications and Technologies*, 2012, pp. 727–732.

[15] H. R. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’06)*, vol. 1, 2005, pp. 695–701.

[16] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, “Opposition-based differential evolution,” *IEEE Transactions on Evolutionary computation*, vol. 12, no. 1, pp. 64–79, 2008.

[17] M. Nogueira and O. Saavedra, “Estratégias evolutivas aplicadas à resolução de otimização multimodal,” in *Simpósio Brasileiro de Automação Inteligente*, 1999.

[18] J. M. Dieterich and B. Hartke, “Empirical review of standard benchmark functions using evolutionary global optimization,” *arXiv preprint arXiv:1207.4318*, 2012.

[19] X. Shan, K. Liu, and P.-L. Sun, “Modified bat algorithm based on lévy flight and opposition based learning,” *Scientific Programming*, vol. 2016, 2016.

[20] B. Zhu, W. Zhu, Z. Liu, Q. Duan, and L. Cao, “A novel quantum-behaved bat algorithm with mean best position directed for numerical optimization,” *Computational intelligence and neuroscience*, vol. 2016, 2016.

[21] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, Dec. 2006.