International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.10 (2016), pp.93-102
http://dx.doi.org/10.14257/ijmue.2016.11.10.08

ILBA: An Improved Bat Algorithm with Inertia Weight Factor and Lévy Flight

Ma weifeng[1], Shi Hao[2] and Sun Xiaoyong[1]

[1]School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
[2]Hangzhou Cropsoft Science & Technology Co., Ltd., Hangzhou 310023, China
E-mail{mawf@zust.edu.cn, constr@163.com, sunxy@zust.edu.cn}

Abstract

Aiming at such shortcomings of traditional Bat algorithm as low precision convergence, premature convergence, slow convergence, an improved BA based on inertia weight factor and Lévy flight (ILBA) has been proposed, which has made two modifications on update equations of bat flying position in BA, using inertia weight factor to keep the flight inertia of bat individual, adaptively adjust the exploitation mechanism of the algorithm in different iteration periods, make the algorithm achieve better convergence precision and altering the strategy about position update of bats from Brownian random walks into Lévy flights strategy to effectively avoid local optimism of the algorithm and guarantee its exploration mechanism while taking advantage of heavy-tailed effect of Lévy flight to speed up the convergence. By means of 4 typical test functions simulation, the results show that ILBA boasts faster convergence and superior optimal performance compared to traditional BA and LBA.

***Keywords**: Bat algorithm, Inertia weight factor, Lévy flight, Heuristic optimization*

1. Introduction

Novel heuristic algorithms have emerged in recent years: the krill herd algorithm (KH) that simulates the social nature of the krill swarms [1], the pigeon-inspired optimization (PIO) that simulates the pigeons returning home [2], and the spider monkey optimization algorithm (SMO) that simulates the search of spider monkey for food [3]. These heuristic algorithms address the complicated non-linear optimization problem by simulating the behaviors of living creatures in nature. However, there is no heuristic algorithm which can solve all optimization problems alone [4-6].

The bat algorithm (BA) [7] is a meta-heuristic algorithm that Yang XinShe proposed in 2010. It is a novel heuristic algorithm that simulates preying of bat in the dark. BA is uncomplicated and easy to implement, outperforming GA and PSO in terms of convergence accuracy and efficiency [8]. The problem with BA, though, is its low convergence accuracy and proneness to prematurity and local optimum. It is very effective in solving the low-dimension problem, but its performance degrades with the increase in dimensionality [8]. The adaptive bat algorithm (ABA) [9] addressed the premature convergence of the original BA by self-adapting the flight speed and direction of individual bats. An evolving bat algorithm (EBA) was proposed in [10], which achieves a balance between the local search and global search abilities of BA by using diverse distribution of herds and the invasive weed optimization algorithm (IWO). It was proposed in [11] to balance the local search and global search abilities by adjusting the loudness and pulse emission rate of individual bats. The Lévy bat algorithm (LBA) in [12] and [13] provided an approach to premature convergence of BA by using the Lévy flight search strategy, but its convergence rate is low and the convergence accuracy is poor.

This paper proposes a novel bat algorithm on inertia weight factor and Lévy flight (ILBA) which improves the equation of bat flights in two ways. First, due to continuity of flight speed among individual bats, the inertia factor similar to that used in the particle swarm optimization (PSO) is utilized to maintain the flight speed of bats. ILBA can self-adapt the local search ability by using the inertia factor, resulting in higher optimization accuracy. Second, the Lévy flight strategy is employed to guide the bats' flight, thereby expanding the search space and preventing the search from obtaining a local optimal solution. Finally, classical simulations based on test functions verify that ILBA outperforms the original BA and LBA in terms of accuracy and convergence rate.

2. Bat Algorithm

2.1. Preying of Bats

BA simulates the bat's preying through echolocation. While hunting for prey in the dark (optimal solution), the bat emits sound bursts from its mouth in flight, hears and collects the echo, and determines the position and size of the object as well as the object's motion. During bat flight, the frequency of the emitted pulse is 10kHz-100kHz, lasting 5-20 milliseconds. When the bat searches for prey, the emitted ultrasonic burst has the highest loudness, which is helpful in transmitting the ultrasonic burst across a long distance. When the bat finds the prey, the loudness of the sound decreases and the pulse frequency increases gradually, which is conducive to accurate positioning of the prey for the bat.

2.2. BA Steps

BA proceeds as follows:

Step 1: Initialize the parameters. Set up the number of bat populations (denoted by n), the position of individual bats (denoted by x_i, i = 1,2,3, …, n), the flight speed of individual bats (denoted by v_i), the loudness of sound A_0, and the pulse emission rate r_0.

Step 2: Compute the fitness function $f(x)$, and store the optimal solution.

Step 3: Randomly generate the pulse frequency, update the flight speed and position.

Let x_i^t denote the position of the individual bat in the d-dimension space, v_i^t denote the flight speed, and f_i denote the pulse frequency. The position x_i^t, speed v_i^t and sounding frequency f_i of the individual bat at time step t are given below:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{1}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

where f_{max} and f_{min} denote the maximum and minimum values of the bat's pulse frequency, $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. In (2), x_* is the current global best location (solution) which is located after comparing all the solutions among all the n bats.

Step 4: Generate a new solution based on the pulse emitting rate r_i.

During local search, once a solution is selected among the currently obtained best solutions, a new solution for each bat is generated locally using random walk.

$$x_{new} = x_{old} + \varepsilon A^t \tag{4}$$

where $\varepsilon \in [-1,1]$ is a random number that follows uniform distribution, A^t denotes the average loudness of all bats at this time step, x_{old} is the current global optimal solution, and x_{new} is the new solution achieved from the current global optimal solution.

Step 5: Accept the new solution based on the loudness A_i and the value of the fitness function. Increase the loudness A_i and decrease the pulse emitting rate r_i.

The formula for updating the loudness A_i and decreasing the pulse generating rate r_i at each iteration is:

$$A_i^{t+1} = \alpha A_i^t \tag{5}$$

$$r_i^{t+1} = r_i^0 [1 - e^{(-\gamma t)}] \tag{6}$$

where A_i^t is the loudness at previous iteration, A_i^{t+1} is the current loudness, α and γ are constant and similar to the cooling factor used for the cooling process in the simulated annealing algorithm [14], r_i^{t+1} is the current pulse emitting rate.

If A_i=0 and r_i=1, BA becomes the standard PSO algorithm; if A_i=r_i is a constant, then BA becomes the harmony search algorithm (HS); for any $\alpha\in$ (0,1) and γ>0, the following assertion holds with an increase in the number of iterations.

$$A_i^t \to 0, \qquad r_i^t \to r_i^0 \quad (t \to \infty) \tag{7}$$

Step 6: Compare the value of fitness among the n bat populations, and find the optimal solution.

Step 7: Repeat Steps (2) to (7), until the condition is satisfied.

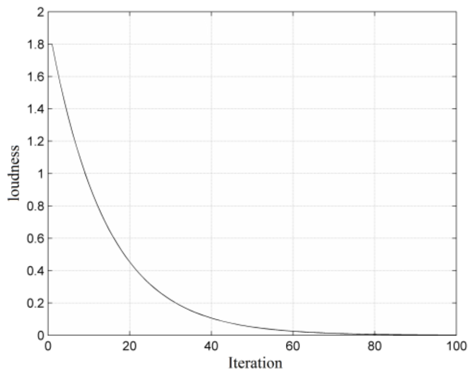

Figure 1. The Changes of Loudness with Iterations

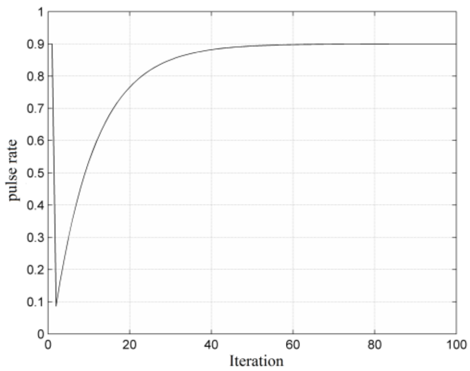

Figure 2. The Changes of Pulse Rate with Iterations

3. BA with Inertia Factor and Lévy Flight Strategy

The traditional BA has issues. For example, its convergence rate is low, the local search ability is better than the global search ability, and it is prone to be trapped in local optimum while solving multi-modal problems [15]. In this paper, the balance between the algorithm's local search ability and the global search ability is achieved by making two modifications to the equation for updating the individual bat's speed.

3.1. Inertia Factor

In the PSO algorithm [16-17], the equation for updating the particle's speed consists of three parts. The first is the inertia regarding the particle's speed. The second is the particle's self-cognition. This part represents the memory of the particle's flight, enables the particles to carry out global search, and prevents particles from being trapped in local optimum. The third pertains to the particle's social cognition. This part represents information sharing among particles and is helpful in improving the global search ability. The inertia factor ω in the first part controls the rate of change of the particle's flight speed. When the value of ω is large, the particle's flight speed varies greatly, the global optimization ability is strong and the local optimization ability is weak, and vice versa.

In BA, the algorithm's global and local search ability is controlled via the pulse emitting rate r. In Step 4 of BA, if $rand>r_i$, the algorithm conducts local search and as the iteration continues, the pulse emitting rate r_i increases. As a result, the algorithm gradually loses its local search ability and the convergence accuracy decreases. The inertia factor ω is utilized to control the position variation range of bat flight, and to balance the algorithm's local and global search ability. As the iteration continues, the value of ω increases, the local search ability improves, and the individual bat approaches the global optimal solution more closely, resulting in faster convergence rate. The position and speed updating formulas (1)-(3) are transformed into (8)-(9).

$$x_i^t = \omega_i \times x_i^{t-1} + v_i^t \tag{8}$$

$$\omega_i = \omega_{min} + \frac{\frac{iter_{max}}{n} - \left\lceil \frac{iter_i}{n} \right\rceil}{\frac{iter_{max}}{n}} (\omega_{max} - \omega_{min}) \tag{9}$$

where ω_i is the inertia factor, ω_{min} is the minimal inertial factor, ω_{max} is the maximum inertia factor, $iter_{max}$ is the algorithm's highest number of iterations, $iter_i$ is the number of iterations, n is the number of populations, $\lceil\ \rceil$ is the rounding off operation.

The inertia factor is defined as:

$$\omega_i = \omega_{min} + \frac{iter_{max} - iter_i}{iter_{max}} (\omega_{max} - \omega_{min}) \tag{10}$$

During the iteration process, the number of iterations ranges from 1 to $iter_{max}$. A small variation range of ω_i results in slower convergence rate and lower convergence accuracy.

3.2. Lévy Flight Strategy

A Lévy flight, named for French mathematician Paul Lévy, is a Markov process. It is a random walk in which the step-lengths have a probability distribution that is heavy-tailed. [18]. The equation of the step length μ and the occurrence frequency $P(\mu)$ is:

$$P(\mu) = \mu^{-\lambda}, \quad 1 < \lambda < 3 \tag{11}$$

According to the research conducted by Viswanathan *et al.* in [19-20], for large-scale search and especially the high-dimension complicated space, expanding the search space by using the Lévy flight strategy can effectively avoid premature convergence and improve the algorithm's convergence speed. An excellent heuristic algorithm should have the ability to prevent premature convergence and provide great optimization performance. In the traditional BA, the parameter β in the frequency updating equation follows the uniform distribution, so the individual bat updates its position based on the Brownian motion. In addition to expanding the individual bat's search space, updating the bat's position through Lévy flight can also prevent it from being trapped in local optimum and improve the bat algorithm's ability in searching the high-dimension space for optimal solution. Unlike the Gaussian and Poisson distributions, the Lévy distribution can make it hard to access the previous position, which is helpful in jumping out of the local optimum. By using the Lévy flight strategy, the position and speed updating formulas (1)-(3) can be written as:

$$x_i^t = x_i^{t-1} + (x_i^{t-1} - x_*) \otimes Levy(\lambda) \tag{12}$$

where $x^{t-}_{1\ i}$ is the spatial location of the ith bat at the (l-1)th search, x_* is the optimal location of the current bat, $Levy\ (\lambda)$ is the random search vector whose step length follows the Lévy distribution, $\otimes$ is the vector operation.

Based on (9) and (12), the equation for updating the bat's position in ILBA is:

$$x_i^t = (\omega_{min} + \frac{\frac{iter_{max}}{n} - \left\lceil \frac{iter_i}{n} \right\rceil}{\frac{iter_{max}}{n}} (\omega_{max} - \omega_{min})) \cdot x_i^{t-1} + (x_i^{t-1} - x_*) \otimes Levy(\lambda) \tag{13}$$

4. Simulations

To evaluate the performance of ILBA, simulations are conducted using Matlab2010b with four standard test functions. Its performance is compared with that of the original BA and LBA [12].

4.1. Standard Test Functions

(a) Sphere

$$f_1(x) = \sum_{i=1}^{n} x_i^2, \quad -10 \le x_i \le 10 \tag{14}$$

This function is a unimodal function, which provides the global minimum f_{min}=0 at $x=(0,0,\ldots,0)$.

(b) Griewank

$$f_2(x) = \frac{1}{4000}\sum_{i=1}^{50} x_{\mathrm{i}}^2 - \prod_{\mathrm{i}=1}^{50} \cos\frac{x_i}{\sqrt{i}} + 1, \quad -600 \le x_i \le 600 \qquad (150$$

This function is an overall aggressive multi-modal function. The global minimum f_{min}=0 can be achieved when $x=(0,0,\ldots,0)$.

(c) Ackley

$$f_3(x) = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{50}\sum_{i=1}^{50} x_i^2}) - \exp(\frac{1}{50}\sum_{i}^{50}\cos(2\pi x_i))$$
$$-30 \le x_i \le 30 \tag{16}$$

This function is a multi-modal function, and the global minimum f_{min}=0 can be achieved when $x=(0,0,\ldots,0)$.

(d) Rastrigrin

$$f_4(x) = \sum_{i=1}^{50}[x_i^2 - 10\cos(2\pi x_i) + 10], \quad -5.12 \le x_i \le 5.12 \tag{17}$$

This is a multi-modal function. It has several local minimal values which curve outwards in a sinusoidal manner, making it prone to be trapped in local optimum. The global minimum f_{min}=0 can be achieved when $x=(0,0,\ldots,0)$.

4.2. Setting the Algorithm Parameters

The initialization parameters of BA, LBA and ILBA, are shown in Table1.

Table 1. Initialization Parameters

Parameters	Populations	Iteration	f_{min}	f_{max}	α/γ	Vioce volumn	ω_{min}	ω_{max}	Pulse rate
Value	20/50	1000	0	2	0.9/0.9	1.5	0.2	0.9	0.5

4.3. Analysis of Experimental Results

BA, LBA and ILBA are each run 1,000 times to compute the algorithms' optimal values, worst values, mean values, and the variance of the optimal values. It is stipulated in the experiment that the algorithm is considered to find the optimal value when the difference between the actual value of the fitness function and the theoretically optimal value of the fitness function is less than 0.01. Experimental results are shown in Table2. In the "success" column, "+" denotes the success in finding the global optimal value, and "-" denotes the failure to find the global optimal value.

Table 2. The comparison of BA, LBA 和 ILBA

Functions	d	Algorithm	Optimal	Worst	Mean	Variance	Success
f_1 Sphere	20	BA	1.3670E+01	6.1638E+01	2.8984E+01	1.2984E+01	-
		LBA	9.3865E+00	5.0456E+01	2.6841E+01	1.0147E+01	-
		ILBA	0	1.4658E-09	1.0321E-10	3.0761E-10	+
	50	BA	3.7061E+01	1.4961E+02	8.6454E+01	3.1535E+01	-
		LBA	1.6045E+01	1.1152E+02	6.2319E+01	2.4253E+01	-
		ILBA	0	3.4278E-08	1.2393E-09	6.2467E-09	+
f_2 Griewank	20	BA	2.3801E+02	6.9037E+02	3.8025E+02	1.4347E+02	-
		LBA	1.8468E+02	2.9966E+02	2.5479E+02	1.3899E+02	-
		ILBA	0	1.4056E-09	7.7713E-11	2.3069E-10	+
	50	BA	4.2881E+02	1.1383E+03	6.4035E+02	1.1073E+02	-
		LBA	3.1959E+02	1.0352E+03	5.4239E+02	1.1814E+02	-
		ILBA	0	1.8005E-09	6.8021E-11	2.6179E-10	+
f_3 Ackley	20	BA	1.8220E+01	5.5545E+01	3.2263E+01	1.3420E+00	-
		LBA	1.5810E+01	4.5423E+01	2.6378E+01	1.5796E+00	-
		ILBA	0	7.7783E-05	7.7029E-06	1.8324E-05	+
	50	BA	1.9912E+01	2.1275E+01	2.0671E+01	1.1188E+00	-
		LBA	1.5353E+01	1.7931E+01	1.6754E+01	1.0285E+00	-
		ILBA	0	1.4662E-04	9.0667E-06	2.1980E-05	+
f_4 Rastrigrin	20	BA	1.8695E+02	3.9455E+02	2.9998E+02	2.6300E+01	-
		LBA	1.0322E+02	5.0015E+02	2.8005E+02	3.4276E+01	-
		ILBA	0	1.8841E-02	1.3811E-03	3.5648E-03	+
	50	BA	3.2990E+02	6.2979E+02	4.7922E+02	6.4370E+01	-
		LBA	1.6340E+02	4.8174E+02	2.9310E+02	6.9347E+01	-
		ILBA	0	9.3474E-02	1.9543E-03	1.3209E-02	+

From Table2, it can be seen that BA provides the lowest convergence accuracy while evaluating the function's convergence performance in the 20- and 50-dimension search space. For LBA, the function f_3 (Ackley) provides good accuracy for different dimensions and the relative error is 2.98%. In comparison, the relative error of other test functions is greater than 40%. This means that the convergence accuracy of LBA decreases with increase in the dimensions. In ILBA, functions f_1 - f_4 converge to 0 (optimal value) at different dimensions, and the variance of the optimal values is less than 10^{-2}. This indicates that the convergence accuracy of ILBA is higher than that of LBA at different dimensions.

To evaluate the iterating performance of the test functions, the maximum number of iterations is set to 5,000 and the basic parameters are shown in Table1. The optimization curves of the test functions are shown in Figures 3-8.

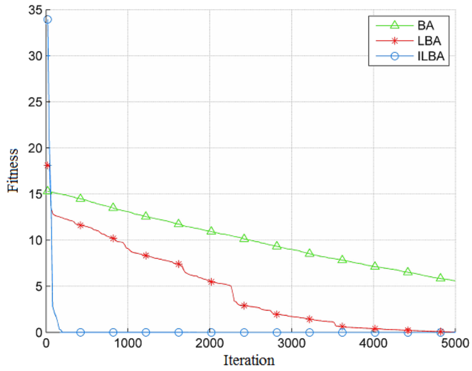

Figure 3. Convergence for 20-D Sphere Function

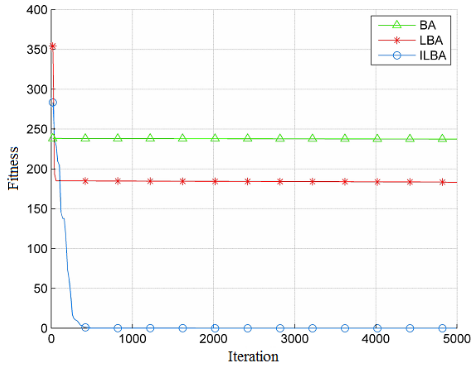

Figure 4. Convergence for 20-D Griewank Function

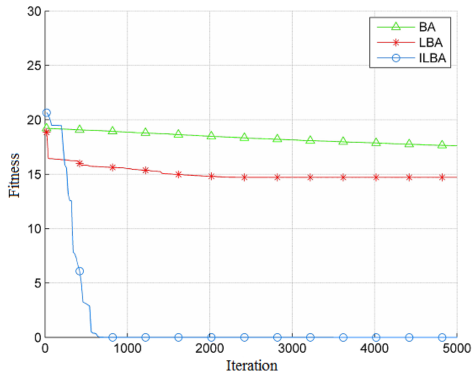

Figure 5. Convergence for 20-D Ackley Function

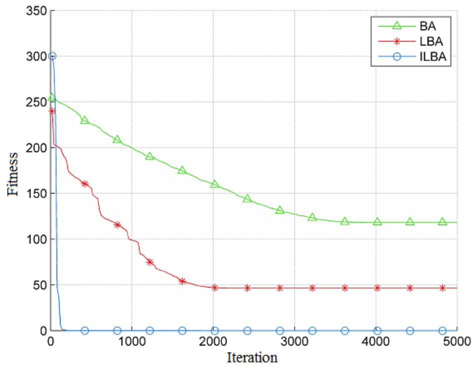

Figure 6. Convergence for 20-D Rastrigrin Function

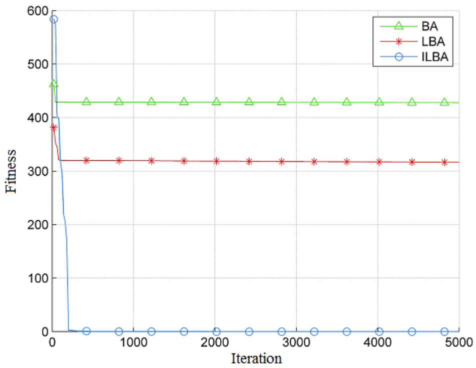

Figure 7. Convergence for 50-D Griewank Function

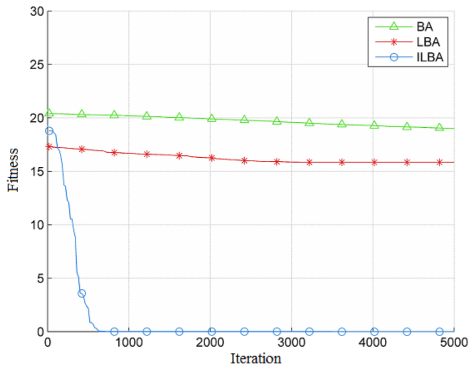

Figure 8. Convergence for 50-D Ackley Function

As shown in Figures 3-8, the convergence value slumps when the Lévy flight strategy is used, which means that the algorithm breaks out of the local optimum during the iteration process. ILBA converges faster than BA and LBA in both high and low dimensions. It can thus be concluded that ILBA outperforms BA and LBA in terms of convergence accuracy and rate.

5. Conclusions

This paper proposes a novel bat algorithm called ILBA to improve the traditional bat algorithm's convergence accuracy and rate. In ILBA, the equation for updating the bat's flight is adjusted through the use of the inertia factor and the Lévy flight strategy.

Simulations based on four standard test functions show that ILBA outperforms BA and LBA in terms of the search ability. Thus, ILBA is suited for solving the high-dimension optimization problem. Since the BA study is in its infancy stage, there are still many issues to be addressed. Moreover, the setting of initial parameters in BA, the functions for loudness and the pulse rate, as well as the combination with other optimization algorithms can be investigated further.

References

[1] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, **(2007)**, pp. 67-82.
[2] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm", Communications in Nonlinear Science and Numerical Simulation, vol. 17, no. 12, **(2012)**, pp. 4831-4845.
[3] H. Duan and P. Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning", Computational Intelligence Systems, vol. 7, no. 1, **(2014)**, pp. 24-37.
[4] J. Bansal, H. Sharma, S. Jadon and M. Clerc, "Spider monkey optimization algorithm for numerical optimization", Mimetic Computing, vol. 6, no. 1, **(2014)**, pp. 31-47.
[5] C. Sur, S. Sharma and A. Shukla, "Egyptian vulture optimization algorithm – a new nature inspired meta-heuristics for knapsack problem", Advances in Intelligent Systems and Computing, Springer, Berlin, Heidelberg, vol. 209, **(2013)**, pp. 227-237.
[6] Y. J. Zheng, "Water wave optimization: A new nature-inspired metaheuristic", Computers & Operations Research, vol. 55, **(2015)**, pp. 1-11.
[7] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", Proceeding of Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Springer Berlin, **(2010)**, pp. 65-74.
[8] I. F. Jr, D. Fister and X. S. Yang, "A hybrid bat algorithm", Elektrotehniski Vestnik, vol. 80, no. 1, **(2013)**, pp. 1-7.
[9] X. W. Wang, W. Wang and Y. Wang, "An Adaptive Bat Algorithm", Lecture Notes in Computer Sciences, Nanning, China, **(2013)**, pp. 216-223.
[10] S. Yilmaz and E. U. Kucuksille, "A new modification approach on bat algorithm for solving optimization problems", Applied Soft Computing, vol. 28, **(2015)**, pp. 259-275.
[11] S. Yilmaz, E. Ugur and Y. Cengiz, "Modified Bat Algorithm", Elektronika IR Elektrotechnika, vol. 20, no. 2, **(2014)**, pp. 71-78.
[12] C. P. Liu and C. M. Ye, "Bat algorithm with the characteristics of Lévy flights", CAAI Transactions on Intelligent Systems, vol. 8, no. 3, **(2013)**, pp. 240-246.
[13] J. Xie, Y. Q. Zhou and H. Chen, "A Bat Algorithm Based on Lévy Flights Trajectory", PR&AI, vol. 26, no. 9, **(2013)**, pp. 829-837.
[14] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing", Science, vol. 220, **(1983)**, pp. 671-680.
[15] G. Wang and L. Guo, "A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization", Applied Mathematics, **(2013)**, pp. 21-25.
[16] J. Kennedy and R. Eberhart, "Particle swarm optimization", proceeding of 6th IEEE International Conference Neural Networks. Perth, Australia, **(1995)**, pp. 1942-1945.
[17] Y. Shi and R. Eberhart, "A modified particle swarm optimize", Proceeding of IEEE World Congress on Computational Intelligence, **(1998)**, pp. 69-73.
[18] G. Mehrdad, Z. Zahra and A. Yazdan, "Computer simulation study of the Lévy flight process", Physica A, vol. 388, **(2009)**, pp. 1509-1514.
[19] G. M. Viswanathan, V. Afanasyev and S. V. Buldyrev, "Lévy flight search patterns of wandering albatrosses", Nature, vol. 381, **(1996)**, pp. 413-415.
[20] G. M. Viswanathana, F. Bartumeus and V. B. Sergey, "Lévy fight random searches in biological phenomena", Physica A, vol. 314, **(2002)**, pp. 208-213.