The Stability Analysis of Bat Algorithm

Janusz P. Paplinski, Miroslaw Lazoryszczak
Department of Computer Architectures and Teleinformatics,
West Pomeranian University of Technology Szczecin,
ul. Zolnierska 52, 71-210 Szczecin, Poland,
janusz.paplinski@zut.edu.pl

Abstract—**The equations describing the position and movement of the new individuals in Bath Algorithm have the form of difference equations. The analysis of the behavior of solutions of this equation and in particular its stability is possible, after omitting of randomness in parameters and treatment of the algorithm as stationary. The simplify analysis of the choice of parameters of the bat algorithm based on the linear stability and behavior of the algorithm is presented in the paper. The study indicates the recommended areas of the location of the parameters, and it shows how the different parameters affect the behavior of the algorithm. A simple tool to speed up the tuning of the algorithm was obtained.**

Keywords—bat algorithms, BA, stability analysis, parameter selection, heuristic algorithms,

I. Introduction

In recent years, the nature-inspired metaheuristic algorithms for optimization problems become very popular. Though these algorithms do not guarantee the optimal solution, they have generally a tendency to find a good solution and become powerful methods for solving many difficult optimization problems [1-3]. The heuristic methods are based on the many different mechanisms occurring in nature. The Genetic Algorithms [4] are based on the biological fundamentals, Tabu Search is based on the social behavior [5], Ant Colony Optimization [6, 7] or Particle Swarm Optimization (PSO) [8] are based on the swarm behavior. The Bats Algorithm (BA) proposed by Yang [9] belongs to these last. The bats use some type of sonar, called echolocation, to detect prey or to avoid obstacles. The echolocation guides their search and allows discriminating different types of insects, even in complete darkness.

There are some much powerful modifications of BA, for instance, BA based on differential operator and Lėvy flight trajectory (DLBA) proposed by Xie at al. [23]. Improved bat algorithm (IBA) proposed by Yilmaz and Kucuksille [24], Enhanced Bat Algorithm proposed also by Yilmaz and Kucuksille [25], and many others, but the simple BA is a base and more popular than other modification. For this reason, the analysis is made for simple BA in the paper.

The main problem for the soft-computing algorithms is a determination of their parameters. The tuning rules are very general and need experiments during a trial and error method. The main idea during the tuning of the parameters is to balance the running algorithm between exploration and exploitation. In the case of too little exploration and intensive exploitation, the algorithm can converge to a local optimum. Otherwise, too much exploration and too little exploitation can give the algorithm with a very small convergence [10 - 12].

The methods from control theory were used in the paper as a useful tool for analyzing a behavior of algorithms. The velocity and the location of the bats are described by difference equations. These equations defined dynamic behavior of the algorithm. The parameters of the algorithm affect on the root locus of the difference equation and its stability, and consequently they are impacting on the speed of convergences.

The rest of the paper is organized as follows. In section 2 the BA algorithm is described. The analysis of the stability of algorithms and some properties of difference equation important for BA is described in section 3. Section 4 contains the example of BA behavior, the experimental results and discussion.

II. Bat Algorithm

The bats have fascinating abilities such as finding their prey and discriminating different types of insects even in complete darkness. Bats use echolocation by emitting high-frequency audio signals and receiving a reflection of those ones. The time delay between emission and detection of the echo and the loudness variation of the echoes allows bats to recognize surrounding. The metaheuristic BA uses some simplicity and idealized rules:

1. All bats use echolocation to appoint a distance and direction to the food. They also can recognize the difference between food/prey and background barriers;
2. The *i*-th bat is at position x_i and flies randomly with a velocity v_i. It emits an audio signal with a variable frequency between $[f_{min}, f_{max}]$, a varying wavelength λ and loudness A_0 in order to search for food. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target;
3. The loudness can vary in many ways. We assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min} Each of the artificial bats has a position vector x_i , velocity vector v_i and frequency vector f_i which is updated during the course of iterations by using the below relations, from (1) to (3). The bats have got

randomly assigned frequency f_i , which is drawn uniformly from the range [f_{min}, f_{max}]:

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

where $\beta \in$ [0, 1] is a random vector with a uniform distribution. The velocity of the bat depends on the position of the current global best solution attained so far x_{bsf}:

$$v_i^t = v_i^{t-1} + \left(x_i^{t-1} - x_{bsf}\right) f_i \tag{2}$$

The new position of the bat follows from his previous position and velocity:

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

The local search procedure is also used. A new solution for a bat is generated locally using current best solution and local random walk:

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

where $\varepsilon \in$ [–1, 1] is a random number with a uniform distribution, while A^t is the average loudness of all bats at t-th time step.

The BA can be considered as a balanced combination of exploration, realized by an algorithm similar to the standard particle swarm optimization and exploitation realized by intensive local search. The balance between these techniques is controlled by the loudness A and emission rate r , updated as follows:

$$A_i\,(t\ +\ 1)\ =\ \alpha A_i\,(t) \tag{5}$$

$$r_i\,(t\ +\ 1)\ =\ r\,A_i\,(0)\,(1 - exp(-\gamma t)), \tag{6}$$

where the coefficients α and γ are constants. In the simplification case $\alpha = \gamma$ is often used. The α can be considered as a similar to the cooling factor in simulated annealing (SA). The loudness and the pulse emission rate are updated only when the new solution is improved. It means that the bat is moving towards the best solution. The pseudocode of BA is presented in Fig.1.

```
1.Initialize the bat population
   xi (i = 1, 2, ..., n) and vi
2. Initialize pulse frequency fi pulse rates
   ri and the loudness Ai
3. While (the stop condition is not
     fulfilled)
4.  Generate new solutions by adjusting
     frequency,
5.  Updating velocities and locations (eq.
     (1) to (3)])
6.  If (rand > ri)
7.   Select a solution among the best
      solutions
8.   Generate a local solution (eq. (4))
9.  End if
10. Generate a new solution by flying
     randomly
11. If (rand < Ai & f(xi) < f(xbsf)
12.  Accept the new solutions
13.  Reduce Ai and Increase ri  (eq.(5, 6))
14. End if
15. Rank bats and find current best xbest
16.End while
```

Fig. 1. The pseudocode of BA

III. Stability Analyses

The very important feature of the soft computing algorithms is their ability to converge to the global optimum. There are some stability analyses of PSO algorithm based on the location of the roots [13-15] and on the Lyapunov function [16-18]. A similar analysis can be made for BA. In order to obtain simple and useful methods for choosing the parameters of BA, the root locus stability is used in the paper.

The equations (2) and (3) can be modified to the form:

$$x_i^t = (1 - g_i)x_i^{t-1} + g_i x_* + v_i^{t-1} \tag{7}$$

where the frequency coefficient $g_i = -f_i$ was adopted only in order to make this equation more readable. For $g_i \in$ [0, 1] the first part of this equation: $(1 - g_i)x_i^{t-1} + g_i \mathrm{x}_*$ is a simply weighted average, between the current position of the bat and the best so far position. It can be treated as a movement into the descent gradient direction. The velocity v_i^t can be treated as a momentum of the flying bat. The intuition behind this strategy is that the current iteration has an inertia, which prohibits sudden changes in the velocity. This is an effective, low-pass filter, i.e., high-frequency changes in the cost function were blocked and greater steps in favorable directions like in the conjugate gradient method are preferred [19-22].

The position of a bat during running the algorithm often does not change from some j-th to j+k-th iteration, and we can write:

$$x_i^t = const \qquad \text{for} \quad i = \langle j, j+k \rangle \tag{8}$$

where j and k are a number of iteration. The best so far position x_{bsf} often does not change, or the changes are negligible, during the j-th to j+k-th iterations. If the sign(f) = const then the velocity of bat v_i (2) growing in one direction systematically searching the search space. The locus of an example individual, its velocity and the locus of the new bat, during algorithm running, are presented in Fig.2. The step 9–10 of the BA in Fig.1, the mutation of the best bat, is visible on the Fig.2c as white samples – the step changes of the value of the position of the new individual.

We can define the value of the velocity of the i-th bat in t-th iteration v_i^t , based on the equation (8):

$$v_i^t = x_i^t - x_i^{t-1} \tag{9}$$

This is some simplification because the velocity of BA is calculated in each iteration, while the bat reaches the new position only in some iteration when the appropriate conditions in line 13 of the algorithm (Fig.1) are met. We simplify this condition and assumed that the new position is every accepted. The assumption allows us to do the analysis of algorithm without making a big mistake. This is equivalent to taking a few samples of the bat with different speeds or more precisely with different frequency and choosing the best one. It can be reduced to a single step, where a value of the velocity is equal to the last used or the best used.

On the base of the relations (2), (3) and (9) the difference equation can be written:

$$x_i^t - (2 + f_i)x_i^{t-1} + x_i^{t-2} = -f_i x_{\mathrm{bsf}} \tag{10}$$

After using Z-transform we obtain the equation in the z-domain:

$$X_i(z) - (2 + f_i)X_i(z)z^{-1} + X_i(z)z^{-2} = -f_i X_{\mathrm{bsf}}(z)z^{-1} \tag{11}$$

The right side of this equation $-f_i X_*(z)z^{-1}$ can be treated as an input signal forced the algorithm. The behaviour of (11) is determined by the characteristic equation:

$$z^2 - (2 + f_i)z + 1 = 0. \tag{1}$$

The roots of this equation are equal:

$$z_{1,2} = \frac{2+f_i \pm \sqrt{f_i^2+4f_i}}{2} \tag{2}$$

The system is stable if the roots lie inside the unit circle in the complex z plane. The roots are complex and $|z_{1,2}| = 1$ for $f_i \in [-4, 0]$. In this case, the roots lie on the stability border. For $f_i < -4$ and $f_i > 0$ the roots have only a real part and at least one of them lies outside the unit circle $|z_{1,2}| > 1$, therefore the algorithm is unstable in this area.

The algorithm exhibits a high oscillatory behavior called ringing when the roots have negative real part, what is for $f_i < -2$. The analysis of root locus allows predicting the behavior of the algorithm and its possibility to achieve the global optimum.

A. The roots on the stability border

The roots are complex and lie on the unit circle for $f_i \in (-4, 0)$. The algorithm works on the verge of stability and behaves as an undamped oscillator. The samples of the responses of the algorithm are presented in Fig.3a. The response of the algorithm is periodic and oscillates around the best individuals. For $f_i < -2$, the ringing occurs and the unit changes of the position of individuals are big. The ringing effect is especially visible in Fig.3a as a high oscillation for $f_i = 3.9$ and $f_i = 3.2$. For $f_i \in (-2, 0)$ and especially for $f_i \to 0$, the changes are smaller and algorithm systematically scans the solution space. For example, for $f = -2$ there is no more than one value between the maximum and minimum, for $f_i = -1.5$ there are one or two intermediate values. The number of intermediate values increases with further growth of f_i. The part of the algorithm responsible for local search is important and allows moving individuals into best solution and decreasing the amplitude of oscillation.

B. The real unstable roots

It is easily calculated from the relation (13) that for $f_i < -4$ or $f_i > 0$ the roots are distinct and real, where one of them lies outside the unit circle. It is the area of an unstable work of the algorithm. The samples of the response of the algorithm for $f_i = [-4.2, -4.1, 0.1, 0.2]$ are presented in Fig. 3b. For the frequency f_i smaller than $f_i < -4$ the response has an oscillatory character with amplitude exponential growing, the faster the farther from the border frequency $f_i = -4$. For the positive frequency $f_i > 0$, the response is aperiodic and growing exponentially, the faster the farther from the $f_i > 0$. For already shown points, lying close to the limit values, the value of the amplitude is near 10^3 after only 15 iterations. This causes that the individuals often going beyond of search space, in the case of constrained optimization. The procedure of repair infeasible individuals becoming important. This procedure is not predefined in the algorithm and strongly depends on the designer of the algorithm. The simply replacing by the limit value is often use. It is always a type of heuristic and it causes that the echolocation does not work properly. The procedure of mutation of the best individuals dominates in that algorithm.

IV. Experiments and Discussion

The population number of BA was equal $N = 40$, the maximum number of function evaluation for each call of BA has been set as 50000. The step 9 of the BA: "Select a solution among the best solutions", is not predefined and can be used any, like a roulette wheel, stochastic universal sampling or other. The first one of above-mentioned methods was used

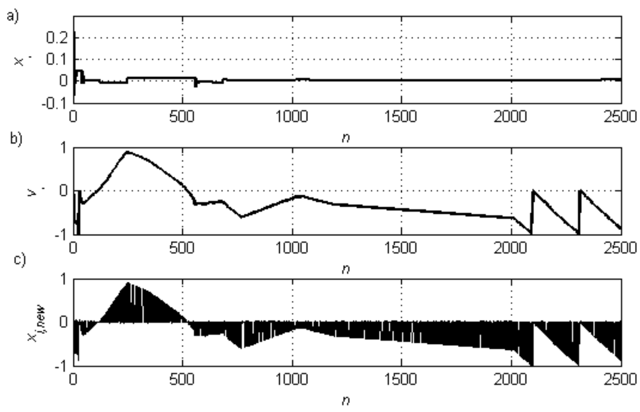


Fig. 2. The example of the behaviour of the bat during running of the algorithm: a) the locus of an individual x_i; b) the velocity of an individual v_i; c) the locus of the new bat $x_{i,new}$; where n – is the number of iteration.

during the experiments. Four benchmark functions, presented in the Table I, were used in order to check and analysis of the efficiency of the BA. They are used as a quality function during searching global maximum value by BA. The BA was run 50 times for each benchmark function. The limitation of search space was presented in the Table I

The BA has three main parameters: the pulse of frequency f_i, the pulse rates r_i and the loudness A_i. The pulse rate r_i was used in line 8 of the pseudocode of BA in Fig.1 and determines how a new bat is calculated: by echolocation or by mutation of the best individual. For a small value of $r_i < 0.5$ the operator of the mutation dominates, otherwise dominates echolocation. The mean values of the performance for the benchmarks, for different pulse rates r_i, are presented in the Table II. The best results were obtained for the small value of r_i equal about 0.2. For this value, the main procedure was a mutation and the echolocation has an auxiliary task. The mutation is used about five times more often than echolocation. The best value of r_i, for Ackley function, is equal 0.6.

a)

b)

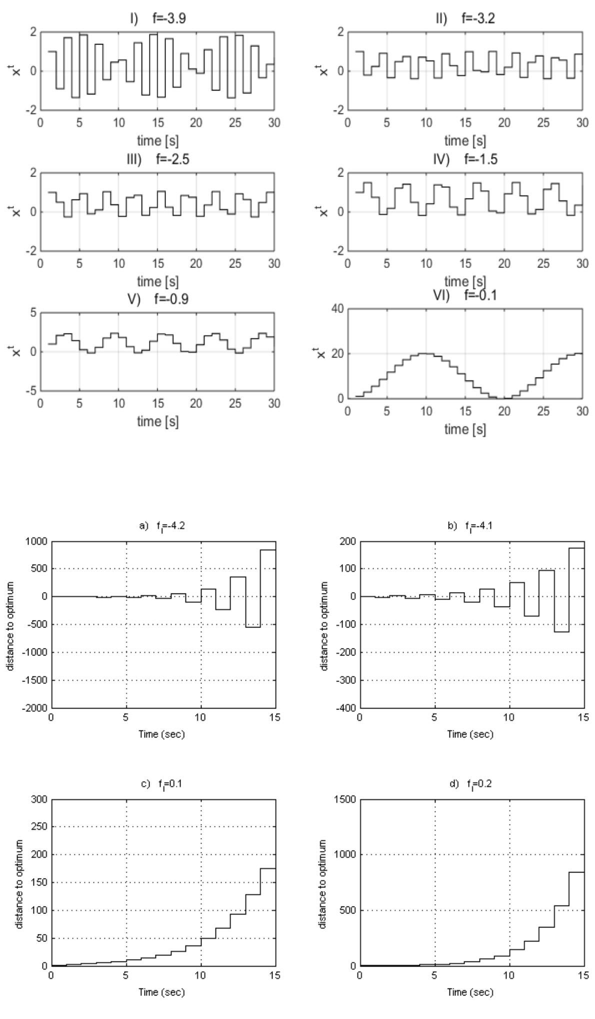


Fig. 3. The sample responses of the algorithm working on a) the stability border; b) the unstable area

The loudness A_i has two tasks; the first one is a determination of a range of mutation, in the second one it is used to check whether to replace a bat by a new individual. For the loudness A_i greater than 1 the second task does not work and the old bat is every replaced. For a small A_i with value near zero, the old bat is never replaced by new one. Of course, it is not a desirable action of the algorithm, therefore the condition of being not too loud (the line 13 in the Fig.1) was ignored in the investigation. The experimental results for the best values of loudness for benchmark functions are presented in Table.1. It is visible that the loudness differs for different functions. The optimal value depends on the nature of the optimized function and can be found experimentally during tuning the algorithm.

TABLE I BENCHMARK FUNCTIONS USED FOR ANALYZING OF THE BA AND THE BEST VALUE OF LOUDNESS AI FOR THE BENCHMARK FUNCTIONS OBTAINED DURING INVESTIGATION

Function	Search range	Type of function	min	the loudness A_i
Sphere	[-5.12, 5.12]	unimod.	0	0.01
Griewank	[-600,600]	multim.	0	4.0
Ackley	[-32,768, 32.768]	multim.	0	0.7
Schwefel	[-500,500]	multim.	0	1.1

a.

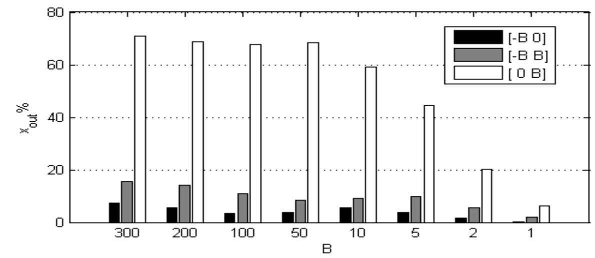
The parameters: pulse rates and loudness do not influence on the stability of algorithm analyzed in section 3. These ones can be treated as parameters of some disturbances, where the pulse rate determines how often disturbances appear and the loudness determines how big they are. More important is the variation of frequency f_i because the difference equation (2) with the variable parameters is nonstationary. The analysis presented in section 3 was made for stationary equation and could not be directly used to nonstationary one. However, this analysis showed a tendency manifested by the algorithm. The main influences of frequency f_i and stability of algorithms can be seen as a number of individuals out of the boundaries of the seeking area. The problem of crossing the border can exist in both, for the location and for the speed. The Fig. 4 presents the percent of new individuals located outside the permitted area, the Fig. 5 shows the percent of new individuals with velocity outside the permissible values, for example functions: Schwefel and Ackley respectively. These Figures present the appropriate value as a function of the lower f_{min} and upper bounds f_{max}. The constraints are divided into three groups presented in Table 3.

A low absolute value of B gives generally less percent of new individuals crossing the bonds. This is expected because for which better results give use a negative frequency (e.g. for Schwefel) or use both negative and positive frequencies (e.g. for Ackley). The first case is a consequence of better fulfilment of stability condition, describe in section 3.

TABLE II. THE MEAN NORMALIZED VALUE OF THE PERFORMANCE, FOR THE BENCHMARKS, FOR DIFFERENT PULSE RATES RI, WITH THE BEST VALUES MARKED

	the emission rate r_i								
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
Sphere	3,57 E+06	**1,00**	1,15	568	1,53 E+04	9,30 E+04	1,36 E+05	1,99 E+05	5,26 E+05
Griewank	1,81	**1,00**	1,08	1,08	1,10	1,09	1,19	1,48	1,53
Ackley	7,67	1,87	1,20	1,29	1,48	1,40	**1,00**	1,06	1,16
Schwefel	**1,01**	**1,00**	1,06	1,04	1,02	1,09	1,17	1,36	1,43

a)



b)

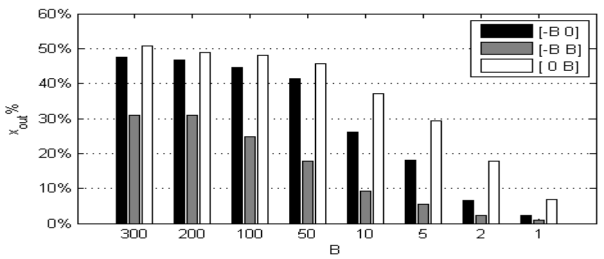
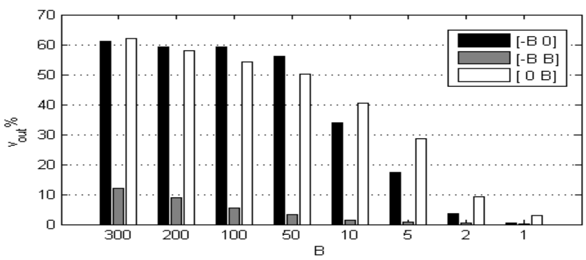


Fig. 5. The percent of individuals outside the permitted area (x_{out}%) as a function of the absolute value of the lower or upper bound of frequency for a) the Schwefell function; b) the Ackley function

a)



b)

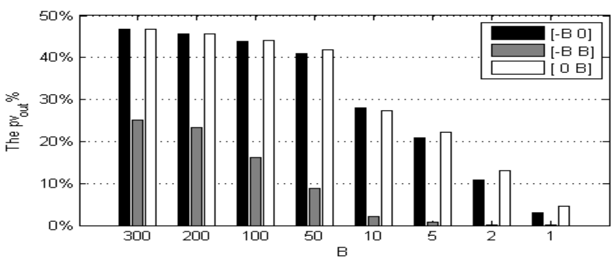
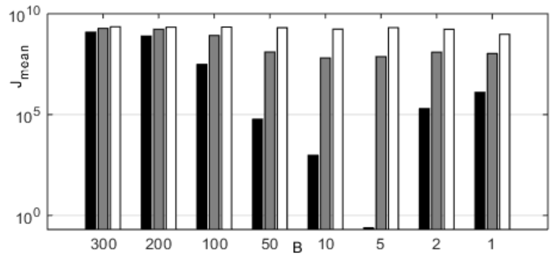


Fig. 6. The percent of individuals with velocity outside the permitted area (v_{out}%) as a function of the absolute value of the lower or upper bound of frequency for a) the Schwefell function, b) the Ackley function.

a)



b)

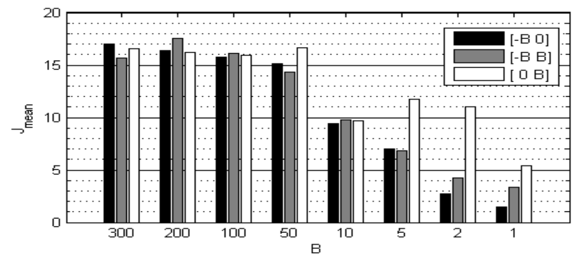


Fig. 4. The mean value of the quality J_{mean} of the algorithm as a function of the absolute value of the lower or upper bound of frequency for a) the Schwefel function, b) the Ackley function.

The second case results from the variation of sign of the frequency f_i and means balancing of positive and negative changes of frequency. This also results from the lower number of new individuals with speed out of the permissible value.

The Fig. 6 present the mean value of quality as a function of the lower f_{min} and upper f_{max} bounds of frequency. The best results are obtained for the negative frequency which fulfills or almost fulfill stability condition. For Schwefel function, the best value of the lower bound of frequency is f_{min} =-5 and is a little out of stability area. It is probably because the frequency is variable and the mean value of frequency is equal 2.5 and lies in the stable area.

TABLE III. THE TYPE OF CONSTRAINTS OF THE FREQUENCY FI AS A FUNCTION OF THE ABSOLUTE VALUE B ($B \geq 0$)

Type of constraints	Lower bound f_{min}	Upper bound f_{max}
I	*-B*	0
II	*-B*	*B*
III	0	*B*

Table IV presents the best mean normalized value of the fitness function for different bounds of frequency obtained for benchmark function. The best solutions are bolded. The Sphere and the Schwefel function are very sensitive to the value of frequency and the stability of BA. The Ackley and Griewank function have the best solution for negative frequency, but the dominance of these frequencies is not big.

V. CONCLUSION

In this paper, the analysis of linear stability of BA based on the root locus was made. It is simplified analysis, because not take into account the non-stationarity of the algorithm, but allows to understand the mechanism of operation of BA and can be used as a simple and useful tool during tuning of BA.

TABLE IV. THE MEAN NORMALIZED VALUE OF THE PERFORMANCE FOR THE BENCHMARKS FUNCTION FOR DIFFERENT BOUNDS OF FREQUENCY

	frequency		
	negative	symmetric	positive
	[-B 0]	[-B B]	[0 B]
Sphera	**1,00**	1,06E+05	1,16E+06
Griewank	**1,00**	1,50	2,26
Ackley	**1,00**	2,33	3,74
Schwefel	**1,00**	2,67E+08	3,96E+09

Base on the experiments made for four benchmark functions, the analysis of the influence of each parameter of BA into its behaviour was made. The best results are obtained for the frequency which fulfills or almost fulfill stability condition. The sensitive of the algorithm on exceeding the stable area are different for each tested functions. It can result from differences in the functions and also from ignoring of non-stationarity in the method..

The pulse emission rate determines the proportions of the use the mutation and echolocation. For the three out of four of the benchmark functions, the most important was mutation.

REFERENCES

[1] M. Seyedmahmoudian, S. Mekhilef, R. Rahmani, R. Yusof, A. A. Shojaei, Maximum power point tracking of partial shaded photovoltaic array using an evolutionary algorithm: A particle swarm optimization technique.Journal of Renewable and Sustainable Energy, *6*(2), 2014, 023102.

[2] M. Amjad, Z. Salam, A review of soft computing methods for harmonics elimination PWM for inverters in renewable energy conversion systems. Renewable and Sustainable Energy Reviews, 33 (2014) 141-153.

[3] R. Talebi, M. M. Ghiasi, H. Talebi, M. Mohammadyian, S. Zendehboudi, M. Arabloo, A. Bahadori, Application of soft computing approaches for modeling saturation pressure of reservoir oils. Journal of Natural Gas Science and Engineering, 20, 2014, 8-15.

[4] M. Gen, R. Cheng, Genetic algorithms and engineering optimization (Vol. 7). John Wiley & Sons, 2000.

[5] F. Glover, E. Taillard, A user's guide to tabu search. *Annals of operations research*, 1993, 41.1: 1-28.

[6] J. P. Paplinski, Continuous ant colony optimization for identification of time delays in the linear plant. In Swarm and Evolutionary Computation, Springer Berlin Heidelberg, 2012, pp. 119-127.

[7] J. P. Paplinski, Time delays identification by means of a hybrid interpolated ant colony optimization and Nelder-Mead algorithm. In Process Control (PC), 2013 International Conference on, IEEE, 2013, pp. 42-46.

[8] J., Kennedy, Particle swarm optimization. In Encyclopedia of Machine Learning (pp. 60-766). Springer US, 2010.

[9] X. S. Yang, A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010), Springer Berlin Heidelberg, 2010, pp. 65-74.

[10] X. S. Yang, Harmony search as a metaheuristic algorithm. In Music-inspired harmony search algorithm, Springer Berlin Heidelberg, 2009, pp. 1-14.

[11] S. S. Hasan, F. Ahmed, Balaning Explorations with Exploitations in the Artificial Bee Colony Algorithm for Numerical Function Optimization. International Journal of Applied Information Systems, 9(1), 2015, 42-48.

[12] F. Xue, Y. Cai, Y. Cao, Z. Cui, F. Li, Optimal parameter settings for bat algorithm. International Journal of Bio-Inspired Computation, *7*(2), 2015, 125-128.

[13] C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection. Information processing letters, *85*(6), 2003, 317-325.

[14] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space. Evolutionary Computation, IEEE Transactions on, 6(1) (2002) 58-73.

[15] J. L. Fernandez-Martinez, E. Garcia-Gonzalo, Stochastic stability analysis of the linear continuous and discrete PSO models. Evolutionary Computation, IEEE Transactions on, *15*(3) , 2011, 405-423.

[16] N. R. Samal, A. Konar, S. Das, A. Abraham, A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 1769-1776, IEEE.

[17] H. M. Emara, A. Fattah, Continuous swarm optimization technique with stability analysis. In American Control Conference, 2004. Proceedings of the ACC 2004 , Vol. 3, 2004, pp. 2811-2817, IEEE.

[18] Q. Jia, Y. Li, The parameter selection of PSO using Lyapunov theory. International Journal of Applied Mathematics and Statistics™, *52*(5), 2014, 69-75.

[19] R. Fletcher, C. M. Reeves, Function minimization by conjugate gradients. The computer journal, 7(2) , 1964, 149-154.

[20] G. Läthén, T. Andersson, R. Lenz, M. Borga, Momentum based optimization methods for level set segmentation. In Scale Space and Variational Methods in Computer Vision, Springer Berlin Heidelberg, 2009, pp. 124-136.

[21] E. Polak, G. Ribière, Note sur la convergence de directions conjugées, Rev. Française Informat. Recherche Opertionelle, 3e année, 16, 1969, 35-43.

[22] B. T. Polyak, The conjugate gradient method in extremal problems.USSR Computational Mathematics and Mathematical Physics, *9*(4), 1969, 94-112

[23] Xie, Y. Zhou, H. Chen, A novel bat algorithm based on differential operator and Lévy flights trajectory. Computational intelligence and neuroscience, 2013.

[24] S. Yılmaz, E. U. Küçüksille, Improved bat algorithm (IBA) on continuous optimization problems. Lecture Notes on Software Engineering,*1*(3), 2013, 279-283.

[25] S. Yılmaz, E. U. Küçüksille, A new modification approach on bat algorithm for solving optimization problems. Applied Soft Computing, *28*, 2015, 259-275.

[26] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989