Uludağ University Journal of The Faculty of Engineering, Vol. 22, No. 2, 2017 RESEARCH

DOI:

PERFORMANCE STUDY OF BAT ALGORITHM AND CLONAL SELECTION ALGORITHM FOR OPTIMIZATION TASKS

Ezgi DENİZ ÜLKER [*]

Received:05.04.2016; revised:12.06.2017; accepted:12.06.2017

Abstract: Evolutionary algorithms are preferred by many researchers in different areas for optimization tasks. It is quite important to find optimum points of problems with less number of iterations. In this paper, performance analysis of two powerful optimization algorithms; bat algorithm and clonal selection algorithm are studied using well-known benchmark functions. The experimental results show that bat algorithm outperforms clonal selection algorithm on most of the selected problems. It is also seen that bat algorithm can produce high quality results even at the first stages of iterations. This paper can be used as guidance of performance comparisons for future studies.

Keywords: Clonal selection algorithm, Bat algorithm, Evolutionary computation, Optimization.

Yarasa Algoritması ve Klonal Seçim Algoritmasının Optimizasyon Problemleri ile Performans Analizi

Öz: Evrimsel algoritmalar, özellikle optimizasyon alanında çalışan bir çok farklı araştırmacı tarafından tercih edilmektedir. Evrimsel algoritmaların verilen problemleri optimize etmenin yanı sıra, bu problemleri az sayıda iterasyon kullanarak çözmeleri bu algoritmalar için önemli bir ayırt edici özelliktir. Bu çalışmada, optimizasyon alanında verimliliği kanıtlanmış iki evrimsel algoritma; yarasa algoritması ve klonal seçim algoritması test fonksiyonları kullanılarak kıyaslanmıştır. Kıyaslama yapılan test fonksiyonlarından elde edilen sonuçlara göre, yarasa algoritması klonal seçim algoritmasına göre daha iyi bir performans göstermiştir. Ayrıca, yarasa algoritması optimizasyonun ilk safhalarında dahi yüksek çözüm kalitesine ulaşmıştır. Bu analiz, gelecek çalışmalar için evrimsel algoritmaların performans kıyaslamaları açısından rehber olarak kullanılabilir niteliktedir.

Anahtar Kelimeler: Klonal seçim algoritması, Yarasa Algoritması, Evrimsel programlama, Optimizasyon.

1. INTRODUCTION

Over the few years, there is an increasing interest to evolution based algorithms and their applications from various fields of research (Adarsh et.al 2016; Sindhuja and Padmavathi, 2016; Vatansever and Şen, 2013). The evolutionary algorithms (EA) mimic the behaviour of nature by selecting the fittest individual and adaptation to the challenging environment. The common attributes of EAs increase the solution quality as well by providing low number of function evalutions.

However, it is seen that no algorithm performs superior for all problems (Wolpert, 1997). Thus, proposing new evolutionary algorithms or modifications, hybridizations of them is still an open area for researchers. Instead of proposing new evolutionary algorithms for problems, modifying the characteristics of the existing ones according to given problems is another preferable method for researchers (Dandy et al., 1996; Gong et al., 2007; Goyal and Patterh,

[*] Computer Engineering Department, Engineering Faculty, European University of Lefke-10, Northern Cyprus. Correspondence Author: Ezgi Deniz Ülker (eulker@eul.edu.tr)

2016). In literature, there exists two powerful evolutionary algorithms that their efficiencies for optimization are proved; bat algorithm Yang (2010) and clonal selection algorithm (De Castro and Von Zuben, 2000). In this paper, we did a performance analysis by using their original characteristics; instead applying any modifications. The main motivation of this work is to analyze their performances on some of the optimization tasks and to observe their weak and strong characteristics. It is also aimed to create a future field of study to use these algorithms' hybrid versions or applying modifications to the observed weak characteristics.

The rest of the paper is organized as follows; Section 2 gives the detailed information of bat algorithm and clonal selection algorithm. Section 3 explains the benchmark functions used in the experiments and the discussions of the results achieved by the algorithms. Lastly, in section 4, the conclusions are given.

2. THE DETAILS OF BAT ALGORITHM AND CLONALG

Bat algorithm (BA) and clonal selection algorithm (CLONALG) are derived from Darwin's evolution theory. Both of them use the common steps of evolution such as selecting the best individuals, updating the population according to environmental changes and elimination of weak individuals to increase the survival rate of populations.

Bat algorithm is first proposed by Yang (2010) with the adaptation echolocation behaviour of bats. Bats are quite good in navigation even in the darkness by using their well developed system called echolocation. They send echo to the objects and from the reflection of that echo, distances, coordinates and even the identification of objects can be determined. The bat algorithm uses this powerful echolocation behaviour of bats. According to BA, each bat in the population moves towards to prey with a velocity (V_i) at position (X_i) with frequency (f), rate (r) and loudness (A). In each iteration, the values are updated for each bat who moves to the prey, until a bat reached to the food source. The steps of BA is given in Figure 1.

Step 1. Initialize the population of bat with variables; velocity (V), position (X), frequency (f), rate (r) and loudness (A).
Step 2. Calculate the fitness value of each bat and rank them.
Step 3. Update the variables V, X, f, r, and A by applying following equations;

$$f_i = f_{min} + (f_{max} - f_{min})\beta$$
$$V_i^{t+1} = V_i^t + (X_i^t - Best)\, f_i$$
$$X_i^{t+1} = X_i^t + V_i^{t+1}$$
$$A_i^{t+1} = \alpha\, A_i^t$$
$$r_i^{t+1} = r_i^0\, [1-\exp^{\gamma t}]$$

Step 4. If ($rand > r_i$)
Select a solution among the best ones.
Generate a solution around the selected solution by using the following equation;

$$X_{new} = X_{old} + \varepsilon\, A_i^t$$

else
Generate a new solution randomly.
Step 5. If ($rand < A_i^t$ & $f(X_i) < f(X_i^*)$)
Accept new solution to the population
Step 6. Calculate the fitness value of each bat.
Step 7. Repeat steps 3-6 until a stopping criterion is met.

Figure 1:
The main steps of Bat Algorithm.

In Figure 1, β is a random number in between (0-1), f_{min} - f_{max} are the boundaries of frequency (0-100), *Best* is the best solution achieved so far by the algorithm in the population, α and ɣ are the constants to update the loudness (A) and rate (r) in between (0-1), ε is a random number in between (0-1), A^t is the average loudness at time t.

CLONALG is first proposed by De Castro and Von Zuben (2000) by adaptation of immune system of organisms. When an intruder called antigen enters to an organism, antibodies are generated to bind them. When the same antigen enters to the organism, antibodies with high

affinity values are generated by recognizing the antigen. The CLONALG uses the idea of antigen-antibody relationship as the steps are explained in Figure 2.

Step 1. Initialize the population of antibodies.
Step 2. Calculate the affinity values of each antibody.
Step 3. Select *n* number of antibodies from the best ones.
Step 4. Clone the antibodies proportionally to their affinity values.
Step 5. Apply mutation to the antibodies according to their affinity values.
Step 6. Calculate affinity values of each antibody.
Step 7. Discard *n* number of the worst antibodies while introducing *n* number of randomly generated antibodies.
Step 8. Repeat Steps 2-7, until a stopping criterion is met.

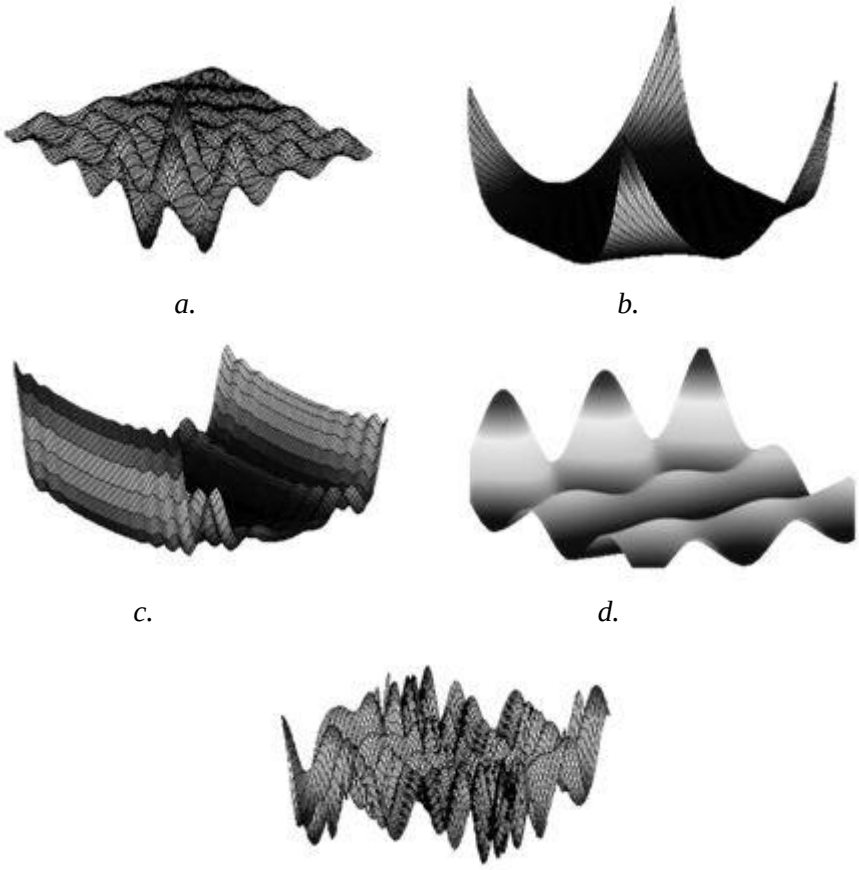
Figure 2:
The main steps of CLONALG.

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, the benchmark functions considered for the performance analysis of BA and CLONALG are presented. The detailed information of benchmarks is given in Table 1 and the graphical representations of benchmarks in 3D are provided in the Figure 3.

Table 1. Selected benchmarks for experiment.

Function name	Function expression	Range	$f(x^*)$	Type
Shubert	$\sum_{i=1}^{5} i\cos\left((i+1)x_1+i\right)\sum_{i=1}^{5} i\cos\left((i+1)x_2+i\right)$	[-10,100]	-186.73	Multimodal
Beale	$(1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1{x_2}^2)^2 + (2.625 - x_1 + x_1{x_2}^3)^2$	[-4.5,4.5]	0	Multimodal
Levy	$sin^2(\pi\omega_1) + \sum_{i=1}^{d-1}(\omega_i - 1)^2\,[1 + 10sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2[1 + sin^2(2\pi\omega_d)]$ $\omega_i = 1 + \frac{x_i - 1}{4}$ $i = 1, \dots, d$	[-10,10]	0	Multimodal
Penalized	$\frac{\pi}{n}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\,[1 + 10sin^2(\pi y_{1+1})] + (y_{n-1})^2\right\} + \sum_{i=1}^{n} u(x_i, a, k, m)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m \;\; if\; x_i > a \\ 0\; if - a \le x_i \le a \\ k(-x_i - a)^m \; if\; x_i < -a \end{cases}$ $a = 10, k = 100, m = 4$	[-50,50]	0	Multimodal
Eggholder	$\sum_{i=1}^{m-1}\left[-(x_{i+1} + 47)sin\sqrt{\left\lvert x_{i+1} + \frac{x_i}{2} + 47\right\rvert} - x_i sin\sqrt{\lvert x_i - (x_{i+1} + 47)\rvert}\right]$	[-512,512]	-959.64	Multimodal



a. *b.*

c. *d.*

e.

Figure 3:
3D representations of selected benchmark functions.
a. *Shubert function* ***b.*** *Beale function* ***c.*** *Levy function*
d. *Penalized function* ***e.*** *Eggholder function*

The graphical representations which clearly show local and global points of the benchmarks are demonstrated as they appear in the following sources; test functions and data sets (2016), generalized penalized function (2016). The functions are selected on the basis of one global optimum point which is located in multiple local optimums. In the performance analysis of optimization algorithms, obtaining the global optimum point among many local points with less number of iterations is a distinguishing characteristic.

The selected benchmark functions are quite effective to estimate the performance of BA and CLONALG algorithms in terms of convergence rate, the number of iterations and robustness. As it can be seen from the part a in Figure 3, the Shubert function has multiple local points. Additionally, some of the local optimums are quite close to the global optimum point which makes the function difficult to solve by an algorithm. In part b, the Beale function is demonstrated. It has one global optimum which is located among four sharp levels. Part c is a representation of Levy function. It has several local optimums and three peak layers formed with multiple local points. Therefore, finding the global optimum of this function is a challenging task by an algorithm.

In part d, the penalized function has three peak layers with multiple local points. The global optimum is located at the point 0 and it is a distinguishing characteristic for an algorithm to find the global optimum by avoiding local optimum points. In part e, eggholder function representation is given. As it can be seen from the figure, it has large number of local optimum

and it is quite difficult to optimize the function. The global optimum point is located at -959.64 and this is covered by many local optimums.

In order to make a fair comparison between the algorithms, the same values are practiced. As it is known, when the number of iterations increases, both of the algorithms can reach to the global optimums of given problems. Since BA and CLONALG are quite powerful algorithms to find the global optimum points of given problems, the maximum number of iterations is fixed to 10000 which is considered as a low number of iterations for an optimization task. The number of population is selected as 100. BA and CLONALG performances are quite dependent to the selection of their control parameters. However, a selected combination of control parameters for a benchmark function may not perform well for another function. Therefore, instead of giving fixed values to the control parameters for all functions, the best combinations are selected for each function through the observational experiments. The performance analysis is handled by using Windows 7 on an intel i5 processor with 4GB RAM using C++ language. The experimental results are given as *Mean* ± *Stdev* format for each function in Table 2.

Table 2. Experimental results of selected algorithms in *Mean*±*Stdev* format.

Function name	BA	CLONALG
Shubert	-186.7309±0	-143.459±57.43504
Beale	1.89E-06±5.98E-06	3.98E-12±1.26E-11
Levy	1.157E-08±3.659E-08	1.24E-21±3.9212E-21
Penalized	1.2733E-08±3.0041E-08	0.010644±0.03193
Eggholder	-905.25976±138.94255	-552.254±318.228

Mean is the averaged values over 40 independent trials and the *Stdev* is the standard deviation of the same 40 runs. For the function Shubert, BA reached to the global optimum point for all values of 40 trials. However, the obtained average value by CLONALG has a large number of deviation which is an indication of low robustness. For the functions Beale and Levy, both algorithms showed the similar performance. BA and CLONALG reached to the global optimum. However, in terms of solution quality which is shown by *Mean* and in terms of robustness which is shown by *Stdev*, the BA algorithm provides better results than CLONALG.

For the Penalized function, BA obtained the global optimum with more robust values and the convergence rate of CLONALG is not as good as BA. Since the eggholder function has large number of local optimum points, 10000 number of iterations may not be enough to optimize it. However, even at this early stage of optimization, we can see that the solution quality of BA is better than CLONALG and the values are more robust, since *Stdev* of BA is lower.

According to the experimental results, the following approaches can be expressed;

i. All of the algorithms are capable to find the optimum points of given benchmark functions.
ii. Especially for Shubert, Penalized and Eggholder functions, BA outperforms CLONALG.
iii. The results obtained for Beale and Levy denote that CLONALG obtains slightly better results than BA.
iv. For all functions, when the mean values found by an algorithm are better than the mean values obtained by the other algorithm, then the same observation is done for the standard deviation values which is an indication of robustness of the algorithm.
v. In general, BA provides better results in terms of solution quality and robustness than CLONALG for the functions used in this experiment set.
vi. In CLONALG, selection of antibodies from the best affinity values gives faster convergence. However, elimination of antibodies and introducing new random

antibodies to the population can cause low convergence especially for low number of iterations (Ulutas and Kulturel-Konak, 2011).

vii. BA combines desired features of some optimization algorithms (Yang, 2010). In addition to this, BA uses control parameters. Thus, good convergence speed obtained especially for low number of iterations (Bin Basir and Binti Ahmad, 2014). However, it is known that BA is a parameter dependent optimization algorithm and without a well parameter tuning, convergence speed is affected.

4. CONCLUSIONS

In this paper, the comparative results of performance analysis of two effective evolutionary algorithms; BA and CLONALG are studied using benchmark functions. According to the experimental results obtained for selected benchmarks, it can be concluded that BA performs better than CLONALG. The selected algorithms proved that they are capable of finding the optimum points of given problems. In order to observe the difference between the performances, low number of iterations is used.

It is seen that BA is a good combination of some optimization algorithms. Hence, the good solution quality can be achieved at the early stages of iterations with a good parameter tuning. In CLONALG, elimination of antibodies and introducing random antibodies to the population can cause low converge at the early stages of iterations.

REFERENCES

1. Adarsh, B. R., Raghunathan, T., Jayabarathi, T., and Yang, X. S. (2016) Economic dispatch using chaotic bat algorithm, *Energy*, *96*, 666-675. doi: 10.1016/j.energy.2015.12.096.

2. Bin Basir, M.A. and Binti Ahmad, F. (2014) Comparison of Swarm Algorithms for Feature Selections/Reductions, *International Journal of Scientific and Engineering Research*, 5, 479-486. doi: 10.1109/ISPACS.2007.4445974.

3. Dandy, G.C., Simpson, A.R., and Murphy L.J. (1996) An improved genetic algorithm for pipe network optimization, *Water Resources Research*, 32, 449-458. doi: 10.1029/95WR02917.

4. De Castro and Von Zuben, F. J. (2000) An evolutionary immune network for data clustering, *In Neural Network, Proceedings Sixth Brazilian Symposium on*, 84-89. doi: 10.1109/SBRN.2000.889718.

5. Gong M, Jiao L, Zhang L and Ma W. (2007) Improved real-valued clonal selection algorithm based on a novel mutation method, *International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2007*, 662-665. doi: 10.1109/ISPACS.2007.4445974.

6. Goyal, S., and Patterh, M. S. (2016) Modified Bat Algorithm for Localization of Wireless Sensor Network, *Wireless Personal Communications*, 86(2), 657-670. doi: 10.1007/s11277-015-2950-9.

7. Generalized penalized function. (2015,June) .Retrieved from http://al-roomi.org/benchmarks/unconstrained/n-dimensions/172-generalized-penalized-function-no-1

8. Test functions and datasets. (2015, January). Retrieved from http://www.sfu.ca/~ssurjano/optimization.html.

9. Sindhuja, L. S., and Padmavathi, G. (2016) Replica Node Detection Using Enhanced Single Hop Detection with Clonal Selection Algorithm in Mobile Wireless Sensor Networks, *Journal of Computer Networks and Communications*. doi: 10.1155/2016/1620343.

10. Ulutas, B.H. and Kulturel-Konak, S. (2011) A review of clonal selection algorithm and its applications, *Artificial Intelligence Review*, 36(2), 117-138.doi:10.1007/s10462-011-9206-1.

11. Vatansever, F. and Şen, D. (2013) Design of PID Controller Simulator based on Genetic Algorithm, *Uludağ University Journal of The Faculty of Engineering,* 18(2), 7-18. doi: 10.17482/uujfe.33406.

12. Wolpert, D.H. and Macready, WG. (1997) No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation,*1, 67-82. doi: 10.1109/4235.585893.

13. Yang X.S. (2010) A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization*, Springer Berlin Heidelberg, *NICSO 2010*, 65-74. doi: 10.1007/978-3-642-12538-6-6.