

Integrating Chaos to Biogeography-Based Optimization Algorithm

Shahrzad Saremi and Seyedali Mirjalili

Abstract—This work integrates chaotic maps into the recently proposed heuristic algorithm called Biogeography-Based Optimization (BBO) algorithm. The effects of 3 different chaotic maps such as Circle, Sine, and Sinusoidal on improving the performance of BBO are investigated in terms of local optima avoidance and convergence speed. The algorithms are benchmarked on four thirty-dimensional test function such as Sphere, Schwefel, Rasterigin, and Griewank. The results prove that chaotic maps (especially Sine map) are able to improve the performance of BBO.

Index Terms—BBO, chaos theory, chaotic maps, evolutionary algorithm.

I. INTRODUCTION

Heuristic optimization algorithms have become very popular among the researchers over the last decade. One of the main reasons is the flexibility of these algorithms in solving different types of problems. The simplicity is also an important factor of the popularity of these algorithms since majority of heuristic algorithms have been inspired by very simple concepts. It is worth mentioning that what makes this field of study different from other fields is a theorem called No Free Lunch theorem [1].

The NFL theorem proved that there is no algorithm for solving all optimization algorithms. This means that an algorithm can outperform other algorithms in some specific problem, but it can be also defeated on some other problems. This theorem is the foundation of many studies in this field which results in the proposal of new algorithms or improving current techniques to solve particular problems every year. There are many heuristic algorithms proposed so far. Some of the best ones are Genetic Algorithm (GA) [2], Particle Swarm Optimization (PSO) [3], Ant Colony Optimization (ACO) [4], and Differential Equation (DE) [5].

Regardless of different approaches to perform optimization, the common fundamental steps among heuristic algorithms are two phases: exploration and exploitation. Exploration refers to the ability of searching the highest possible regions of a search space, whereas exploitation is the convergence toward the best promising solutions obtained in the exploration phase. There are many approaches to improve these phases such as random walks, local searches, and gradient descents. The literature shows

that chaotic maps are able to successfully improve the performance of heuristic algorithms with very cheap computational burden.

Chaos theory refers to the study of chaotic dynamical systems. Chaotic systems are nonlinear dynamical systems that are highly sensible to their initial conditions. In other words, small changes in initial conditions result in high changes in the final outcome of system. One might think that chaos systems behave randomly, but a system does not necessarily need randomness for providing chaos behavior [6]. In other words, deterministic systems are also able to show chaos behaviors. Recently, these characteristics have been utilized for improving the performance of heuristic optimization algorithms.

In 2009, 12 chaotic maps were applied to PSO by Atlas *et al.*, [7]. They also showed that chaos is able to improve the performance of Bee Colony Algorithm (BCA) [8] as well. In 2012, Gandomi *et al.*, proposed a chaos-enhanced version of accelerated particle swarm optimization [9]. Some of other chaos-enhanced heuristic algorithms are chaotic Genetic Algorithms [10], chaotic Differential Evolution [11], chaotic Simulated Annealing [12], and chaotic Firefly Algorithm [13], [14]. The result of these studies evidences the successfulness of chaos in improving the performance of heuristic algorithms.

One of the most recent algorithms is Biogeography-Based Optimization (BBO) algorithm [15]. It has been proven that the BBO algorithm is able to provide competitive results compared to PSO, GA, ACO, and ES. In this study utilizes 3 chaotic maps are employed to improve the performance of BBO for the first time

II. BIOGEOGRAPHY-BASED OPTIMIZATION ALGORITHM

BBO has been inspired by biogeography [15]. Biogeography refers to the study of organisms based on their locations. The migration and mutation of different species are investigated in this field of study over different geographically separated ecosystems. The main inspiration of the BBO algorithm was from the evolution and balance of preys and predators in different ecosystems. So, the BBO algorithm is considered as an evolutionary algorithm.

In the BBO algorithm the variables of a problem are called habitants and represented as a vector named habitat. Habitants and habitats are similar to genes and chromosomes in GA, respectively. Each habitat is evaluated based on a Habitat Suitability Index (HSI). The HSI is defined by the fitness function, so a fitter habitat is the one with the better fitness value.

The habitats are evolved based on three rules over a pre-defined number of generations. First, habitants tend to

Manuscript received April 9, 2013; revised June 12, 2013.

S. Saremi is with the University of Queensland, St Lucia, and 4072QLD (e-mail: shahrzad.saremi@uqconnect.edu.au).

S. Mirjalili is with the Griffith University, 170 Kessels Rd, and Nathan 4111QLD (e-mail: seyedali.mirjalili@griffithuni.edu.au).

emigrate from high-HSI habitats to low-HSI habitats. Second, low- HSI habitats tend to immigrate habitants from high-HSI habitats. Finally, habitants are manipulated randomly by mutation operator(s).

In the BBO algorithm the search process starts with generating a random number of habitats. Each habitat is assigned immigration, emigration, and mutation rates which simulate the characteristics of different ecosystems. Over the course of iterations, each habitat send/receive habitants to/from different habitats based on immigration and emigration rates. A random number of habitants is also mutated occasionally. Finally, the algorithm is terminated by the satisfaction of an end criterion.

The emigration (μ_k) and immigration (λ_k) of each habitat are formulated as follows:

$$\mu_k = \frac{E \times n}{N} \quad (1)$$

$$\lambda_k = I \times \frac{1-n}{N} \quad (2)$$

where n is the habitant count, N is the maximum number of habitants, E is the maximum emigration rate, and I indicates the maximum immigration rate.

The mutation is defined as follows:

$$m_n = M \times \left(1 - \frac{p_n}{p_{max}}\right) \quad (3)$$

where M is an initial value for mutation defined by user, p_n is the mutation probability of n th habitant, and $p_{max} = \arg \max(p_n), n = 1, 2, \dots, N$.

These immigration and emigration rates are depicted in Fig. 1. This figure shows that high number of habitants corresponds to high probability of emigration and low probability of immigration. Note that mutation keeps habitats as diverse as possible.

It has been proved that the BBO algorithm is able to outperform some of the well-known heuristic algorithms such as PSO, GA, ACO, and ES on fourteen benchmark functions and a real problem [15]. So, the BBO algorithm is considered as a competitive algorithm in the field of optimization. In the following sections the method of improving the performance of BBO utilizing chaotic maps is first proposed. Then, four benchmark functions are employed to examine the proposed approach.

III. CHAOTIC MAPS FOR BBO

In this section the utilized chaotic maps are first presented. Then, the method of integrating these maps into BBO is proposed. We have chosen 3 chaotic maps such as Circle, Sine, and Sinusoidal as shown in Table I and Fig. 2.

Generally, one of the methods of utilizing chaos in heuristic algorithms is to use them instead of random values. In other words, chaotic maps are substituted by random values to provide chaotic behaviors for heuristic algorithms. So, we employ chaotic maps whenever there is a need for a random value (possibly for defining the probability values) in the interval of $[0, 1]$. In the BBO algorithm, the most important random values are calculated to choose a habitat for emigrating new habitants as shown in Fig. 3.

In this work the *rand* value is substituted by chaotic maps as follows:

If $C(t) < \lambda_i$

Emigrate habitants from H_i to H_j chosen with the probability proportional to μ_i (4)

End

where $C(t)$ is the value of chaotic maps in the t -th iteration.

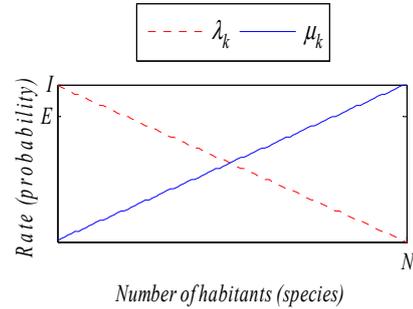


Fig. 1. Emigration (μ_k) and immigration (λ_k) rates.

TABLE I: CHAOTIC MAPS

Name	Chaotic map	Range
Circle [16]	$x_{i+1} = \text{mod}\left(x_i + b - \left(\frac{a}{2\pi}\right) \sin(2\pi x_i), 1\right), a = 0.5$ and $b = 0.2$	(0,1)
Sine [17]	$x_{i+1} = \frac{a}{4} \sin(\pi x_i), a=4$	(0,1)
Sinusoidal [18]	$x_{i+1} = ax_i^2 \sin(\pi x_i), a=2.3$	(0,1)

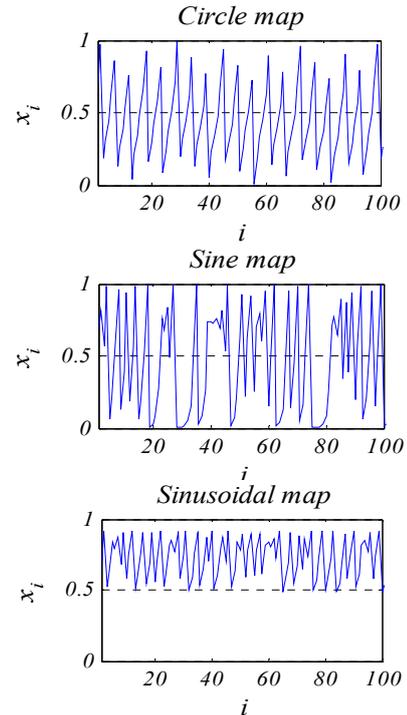


Fig. 2. Utilized chaotic maps.

Select H_i with the probability proportional to λ_i ;

If $\text{rand} < \lambda_i$

Emigrate habitants from H_i to H_j chosen with the probability proportional to μ_i

End

Fig. 3. Pseudo code of immigrating mechanism.

The BBO algorithm starts with a random population. The habitants chosen by the chaotic probability emigrate habitants

to other habitats. In other words chaotic maps are responsible for choosing the origin of immigration rate in our proposed method.

IV. RESULTS AND DISCUSSION

In this section the proposed method benchmarked. We employ 4 thirty-dimensional test functions [19] as shown in Table II. The two-dimensional versions of these functions are depicted in Fig. 4.

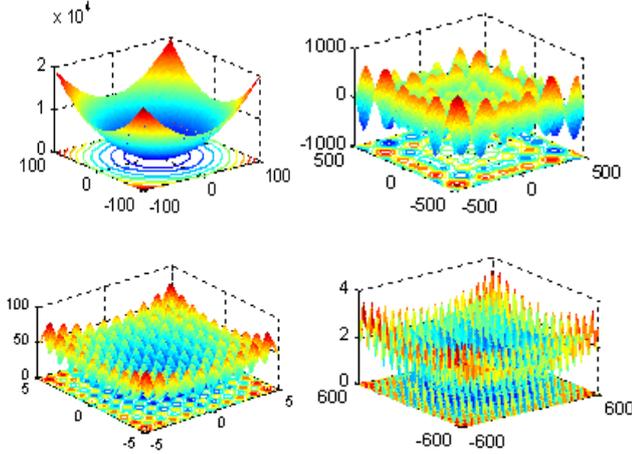


Fig. 4. Two-dimensional representation of utilized benchmark functions.

We apply all chaotic maps to BBO and name them BBO1 (Sphere), BBO2 (Sine), and BBO3 (Sinusoidal). The algorithms were run 10 times. The average (*ave*) and standard deviation (*std*) of these 10 runs are reported in Table III. These metrics have been utilized to examine the ability of algorithms in terms of local optima avoidance.

As can be seen in Table III, BBO1 and BBO2 provide much better results than BBO and BBO3. The reason of poor performance for Sinusoidal map is that this algorithm usually provides values greater than 0.5 as can be seen in Fig. 2. According to equation (4), therefore, there is less immigration probability in case of using Sinusoidal map. In other words, this map weakened exploration which results in poor performance of BBO3.

In contrast, BBO1 and BBO2 provide diverse probabilities over the course of iteration. This result in having different immigration probability rates, so each iteration might emphasize either exploration or exploitation. The results of *ave* and *std* show that these algorithms have better local optimum avoidance compared to BBO and BBO3. The reason of better performance of BBO2 in comparison with BBO1 is that Sine map has more chaotic behavior than Circle map as can be seen in Fig. 2. This helps BBO2 to avoid local minima by having diverse probability of immigration for habitats.

In order to investigate the convergence behavior of the algorithms, the convergence curves are illustrated in Fig. 5. It can be inferred from this figure that BBO2 show the fastest convergence speed, followed by BBO1. This proves that chaotic maps also improve exploitation capability of BBO. The reason is that chaotic maps (especially Sine map) provide different probability values for immigration, so it causes a

good balance between exploration and exploitation.

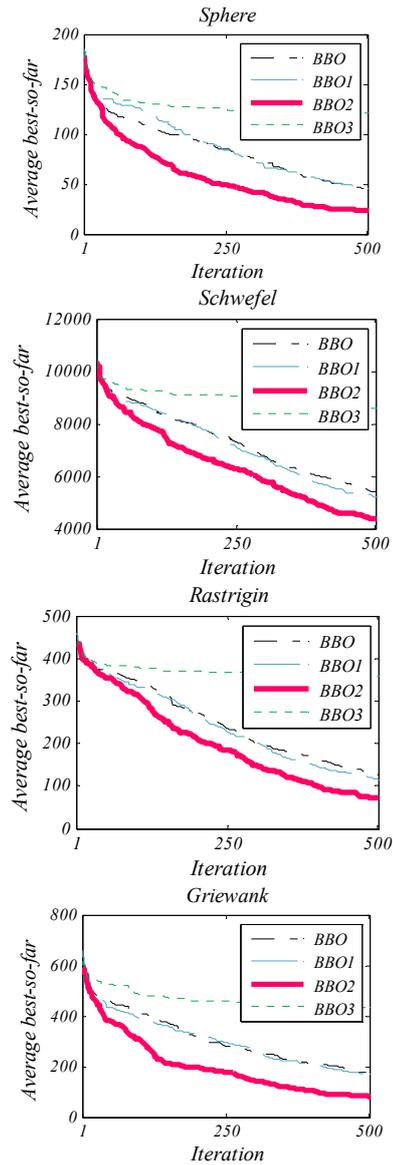


Fig. 5. Convergence curves.

TABLE II: BENCHMARK FUNCTIONS

	Function	Dim ^a	Range
F ₁ : Sphere	$\sum_{i=1}^n x_i^2$	30	[-100,100]
F ₂ : Schwefel	$\sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]
F ₃ : Rastrigin	$\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]
F ₄ : Griewank	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]

^a Indicates the dimension of the function (the number of variables)

In those iterations that the probability of immigration is low, the habitats are less prone to face immigration (sudden changes in habitats), so the mutation operator plays a key role resulting in emphasizing exploitation. In those iterations that the probability of immigration is high, habitats are more prone to immigrate (abrupt changes in habitats) which cause higher exploration.

TABLE III: RESULTS

F	BBO		BBO1		BBO2		BBO3	
	ave	std ^a	ave	std	ave	std	ave	std
F ₁	44.51	13.90	45.65	12.54	23.20	7.782	121.3	8.470
F ₂	5412	488.4	5196	667.2	4379	778.3	8601	494.7
F ₃	125.9	45.11	114.3	22.23	72.46	18.66	357.5	15.41
F ₄	174.6	53.07	170.9	34.54	81.72	19.81	432.8	32.57

^a Standard deviation

To sum up, the results show that Sine maps is able to improve the performance of BBO in terms of not only avoiding local minima but also convergence speed.

V. CONCLUSION

In this work chaos theory was combined to the BBO algorithm for the first time. Three chaotic maps such as Circle, Sine, and Sinusoidal were chosen to be integrated to BBO. A set consists of four test functions benchmarked the performance of the algorithms. The chaotic maps were employed to define the probability of choosing those habitats that should involve in the immigration process. The results proved that the chaotic maps (especially Sine map) are able to improve the performance of the BBO algorithm within the proposed approach in terms of avoiding local optima and convergence speed. Sine map showed the best results because of having more chaotic behavior. This helps BBO2 to avoid local minima by having diverse probability of immigration for habitats. Moreover, the results showed that Sine map causes a good balance between exploration and exploitation.

REFERENCES

- [1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, pp. 67-82, 1997.
- [2] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, pp. 66-72, 1992.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.
- [4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *Computational Intelligence Magazine, IEEE*, vol. 1, pp. 28-39, 2006.
- [5] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341-359, 1997.
- [6] S. H. Kellert, *In the wake of chaos: Unpredictable order in dynamical systems*, University of Chicago Press, 1993.

- [7] B. Alatas, E. Akin, and A. B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons and Fractals*, vol. 40, pp. 1715-1734, 2009.
- [8] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37, pp. 5682-5687, 2010.
- [9] A. H. Gandomi, G. J. Yun, X. S. Yang, and S. Talatahari, "Chaos-Enhanced Accelerated Particle Swarm Optimization," *Communications in Nonlinear Science and Numerical Simulation*, 2012.
- [10] J. Yao, C. Mei, X. Peng, Z. Hu, and J. Hu, "A new optimization Approach-Chaos genetic algorithm," *Systems Engineering*, vol. 1, p. 015, 2001.
- [11] G. Zhenyu, C. Bo, Y. Min, and C. Binggang, "Self-adaptive chaos differential evolution," *Advances in natural computation*, pp. 972-975, 2006.
- [12] J. Mingjun and T. Huanwen, "Application of chaos in simulated annealing," *Chaos, Solitons & Fractals*, vol. 21, pp. 933-941, 2004.
- [13] A. Gandomi, X. S. Yang, S. Talatahari, and A. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, 2012.
- [14] L. S. Coelho and V. C. Mariani, "Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning," *Computers & Mathematics with Applications*, 2012.
- [15] D. Simon, "Biogeography-based optimization," *Evolutionary Computation, IEEE Transactions*, vol. 12, pp. 702-713, 2008.
- [16] W. M. Zheng, "Kneading plane of the circle map," *Chaos, Solitons & Fractals*, vol. 4, pp. 1221-1233, 1994.
- [17] R. L. Devaney, "An introduction to chaotic dynamical systems," 2003.
- [18] Y. Li, S. Deng, and D. Xiao, "A novel Hash algorithm construction based on chaotic neural network," *Neural Computing & Applications*, vol. 20, pp. 133-141, 2011.
- [19] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSO-GSA algorithm for function optimization," in *Proc. 2010 International Conference on Computer and Information Application (ICCIA)*, 2010, pp. 374-377.



S. Saremi was born in Mashhad, Iran on September 21, 1986. In 2011, she received B. Sc degree in information technology (multi media) from Multi Media University, Malaysia. She is currently doing master of interaction design at the University of Queensland, Australia. Her main research interests include Augmented Reality, Education, and optimization.



S. Mirjalili was born in Yazd, Iran on April 21, 1986. In 2008, he received B. Sc degree in computer Engineering (software) from Yazd University, Iran. He obtained M. Sc degree in computer science from Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia, in 2011. His research interests include Multi-objective Optimization, Robust Optimization, Evolutionary Algorithms, and Artificial Neural Networks.