Chaotic Elephant Herding Optimization Algorithm

Eva Tuba
Faculty of Informatics and Computing
Singidunum University
Belgrade, Serbia
etuba@ieee.org

Romana Capor-Hrosik
Institute for Marine and Coastal Research
University of Dubrovnik
Dubrovnik, Croatia
rcapor@unidu.hr

Adis Alihodzic
Faculty of Mathematics
University of Sarajevo
Sarajevo, BiH
adis.alihodzic@pmf.unsa.ba

Raka Jovanovic
Qatar Environment and Energy Research Institute
Hamad Bin Khalifa University
Doha, Qatar
rjovanovic@hbku.edu.qa

Milan Tuba
Department of Mathematical Sciences
State University of Novi Pazar
Novi Pazar, Serbia
tuba@ieee.org

***Abstract*—Swarm intelligence algorithms represent stochastic optimization algorithms that proved to be powerful for finding suboptimal solutions for hard optimization problems. Elephant herding optimization algorithm is a rather new and promising representative of that class of optimization algorithms that has already been used in numerous applications. In recent years, chaotic maps were incorporated into the swarm intelligence algorithms in order to improve the search quality. In this paper we introduced two different chaotic maps into the original elephant herding optimization algorithm. The proposed methods were tested on 15 benchmark functions from CEC 2013. Obtained results were compared to the regular elephant herding optimization algorithm as well to the particle swarm optimization. Test results proved that the proposed chaotic elephant herding optimization algorithm has better performance and obtained better results.**

I. Introduction

Optimization problems are often a part of numerous real life applications. In most cases, practical tasks and problems that need to be solved can be represented as an optimization problem. Optimization problems can be defined as the pair (S, f), where S is the set of possible solutions (search space), while f is the objective function. The goal is to find the global optimum $s* \in S$, a solution that has better objective function value than all other solutions in the search space.

Real life optimization problems are usually highly nonlinear hard optimization problems with numerous local optima. For such optimization problem, usually there are no deterministic algorithms that can find the optimal solution in reasonable time due to computational complexity. For finding a solution for the hard optimization problems some heuristics or metaheuristics are often used. Nature inspired algorithms, especially swarm intelligence algorithms, represent one class of metaheuristics that are widely and successfully used.

Swarm intelligence algorithms mimic swarm behavior from the nature. They contain a swarm of individuals that follow rather simple rules and exchange information between themselves. These simple agents collectively exhibit remarkable intelligence which is used for creating metaheuristics for solving the optimization problems.

This research is supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, Grant No. III-44006.

Swarm intelligence algorithms mimic swarms from the nature such as ant or bee colonies, and certain swarming behaviors as animal herding, food harvesting, nesting. Particle swarm optimization (PSO) and ant colony optimization (ACO) are among the oldest swarm intelligence algorithms. Nowadays, various swarm intelligence algorithms have been proposed such as firefly algorithm [1], [2], fireworks algorithm [3], [4], artificial bee colony [5], etc.Swarm intelligence algorithms were applied to numerous problems such as the traveling salesman problem [6], WSN node localization [7], support vector machine optimization [8], [9], [10], image registration [11], portfolio optimization problem [12], etc.

Optimization algorithms have been constantly improved. One of the latest modifications that is frequently used to improve the search ability is based on the chaos theory. Various chaotic maps such as circle, Gauss/mouse, sine and others were used, rather then random values, in swarm intelligence algorithms [13]. In [14] parameters of the bat algorithm were replaced by different chaotic maps and the results were compared to the original bat algorithm. It has been shown that some chaotic maps found better solutions compared to the original algorithm. In [15], Chebyshev map was introduced into the fruit fly optimization algorithm and experimental results proved superiority and reliability of the chaotic fruit fly algorithm. Ten one-dimensional chaotic maps were implemented and compared for fireworks algorithm in [16]. In [17], tent and logistic maps along with Gaussian mutation were introduced to the PSO. The proposed PSO achieved better results comparing to the standard PSO. Besides these, chaos based cuckoo search [18], chaos based harmony search [19], and others were proposed and applied to different problems [20], [21].

In this paper we propose chaos based elephant herding algorithm (CEHO). Elephant herding algorithm (EHO) is

978-1-5386-4772-1/18/$31.00 ©2018 IEEE

a recent swarm intelligence algorithm that was applied to various problems. We used two one-dimensional chaotic maps and tested the proposed algorithms on standard benchmark functions from CEC 2013. The results were compared to the original EHO and the standard PSO.

The rest of the paper is organized as follows. Elephant herding optimization algorithm, along with the used chaotic maps, are described in Section II. In Section III we present simulation results, along with the comparison to the other algorithms. Conclusion and suggestion for further work are given in Section IV.

II. Chaotic Elephant Herding Optimization Algorithm

Elephant herding optimization algorithm (EHO) represents very recent swarm intelligence algorithm. It was proposed by Wang et. al. in 2016 [22]. Simplified elephant herding behavior was used to implement the EHO algorithm. Even though it is a rather new metaheuristic, EHO has already been successfully applied to numerous problems such as multilevel thresholding [23], support vector machine parameter tuning [24], scheduling problem [25], etc.

In general, all swarm intelligence algorithms contain the global and local search. Global search, or exploration, is used to widely investigate search space in order to prevent getting stuck in local optima. On the other hand, local search, or exploitation, tries to find better solution in smaller promising areas of the search space. With this in mind, elephants herding behavior was described as follows. One elephant population divides into several clans. Each clan follows matriarch, i.e. it has female leading elephant. In each generation, certain number of male elephants leaves the clans to live a lonely life far away from the clan. In terms of swarm intelligence algorithm, clans represents local search, while male elephants that leave the clan implement global search. Matriarch is the solution (elephant) that has the best fitness function value in the clan. On the other hand, male elephants that move are solutions with the worst fitness function values.

Elephant herding optimization algorithm is described as follows. Elephant population is divided into k clans. Initially, D-dimensional solutions are generated randomly in the search space with lower bound x_{min} and upper bound x_{max} by the following equation:

$$x = x_{min} + (x_{max} - x_{min} + 1)rand \tag{1}$$

where $rand$ is random number form uniform distribution in the range $[0, 1]$.

In each generation, solutions are changing in the following way. Member j of the clan i moves influenced by the solution $x_{best,ci}$ with the best fitness function value in the clan c_i [22]:

$$x_{new,ci,j} = x_{ci,j} + \alpha(x_{best,ci} - x_{ci,j})rand \tag{2}$$

where $x_{new,ci,j}$ is the new solution j in clan c_i while $x_{ci,j}$ represents the solution in previous generation, parameter $\alpha \in [0, 1]$ is the algorithm parameter that needs to be set according to the considered problem, while $rand \in [0, 1]$ is a random number from uniform distribution. A scale factor α determines the influence of the best solution.

Position of the best solution in each clan is updated according to the following equation [22]:

$$x_{new,ci} = \beta x_{center,ci} \tag{3}$$

where $\beta \in [0, 1]$ represents the second algorithm parameter and it controls the influence of the clan center $x_{center,ci}$. Clan center is defined as:

$$x_{center,ci,d} = \frac{1}{n_{ci}} \sum_{l=1}^{n_{ci}} x_{ci,l,d} \tag{4}$$

where $1 \le d \le D$ represents the d^{th} dimension and n_{ci} is the number of solutions in clan ci.

In each generation, exploration is done in the following way. In each clan, m_{ci} solutions with the worst fitness values of the clan ci are chosen to be replaced by the following solutions:

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1)rand \tag{5}$$

where x_{min} and x_{max} represent lower and upper bounds of the search space, respectively. Parameter $rand \in [0, 1]$ again represents a random number form the uniform distribution.

EHO algorithm is summarized in the pseudo code presented in Algorithm 1.

Algorithm 1 Pseudo-code of the EHO algorithm

```
Initialization
Set generation counter t=1, set maximum generation MaxGen
Initialize the population
repeat
   Sort all the elephants according to their fitness
   for all clans c_i in the population do
      for all elephants j in the clan c_i do
         Update x_{ci,j} and generate x_{new,ci,j} by Eq. (2)
         if x_{ci,j}=x_{best,ci} then
            Update x_{ci,j} and generate x_{new,ci,j} by Eq. (3)
         end if
      end for
   end for
   for all clans c_i in the population do
      Replace the worst elephant in clan c_i by Eq. (5)
   end for
   Evaluate population by the newly updated positions
until t < MaxGen
return the best found solution
```

A. Chaotic Maps

Swarm intelligence algorithms have been proved to be very powerful for finding relatively good solutions fore various hard optimization problems. Because of that, further improving of these algorithms became an active research topic. One of the rather often used improvements is replacing random values by chaotic maps in swarm intelligence algorithms. Due to the fact that chaotic maps generate numbers with non-repetition and ergodicity, improved search can be expected [19]. In this paper we considered two different one-dimensional maps: circle and sinusoidal map.

Circle map is defined by the following equation:

$$x_{k+1} = \left[x_k + b - \frac{a}{2\pi}\sin(2\pi x_k)\right] \mod 1 \qquad (6)$$

where for $a = 0.5$ and $b = 0.2$ the generated chaotic sequence is within (0, 1).

Sinusoidal map is defined as:

$$x_{k+1} = a x_k^2 \sin(\pi x_k) \qquad (7)$$

where for $a = 2.3$ and $x_0 = 0.7$ the following simplified form can be used:

$$x_{k+1} = \sin(\pi x_k) \qquad (8)$$

The proposed chaotic maps were used to generate chaos sequence of numbers. We replaced random numbers in Eq. 1, Eq. 2 and Eq. 5 by the numbers from the chaos sequences.

III. Simulation Results

Chaotic based EHO algorithms were implemented in Matlab R2016a and simulations were performed on the platform with Intel ® Core^{TM} i7-3770K CPU at 4GHz, 8GB RAM, Windows 10 Professional OS.

The proposed algorithms were tested on 15 well-known benchmark functions proposed in CEC 2013. We tested on 5 unimodal and 10 multimodal 10-dimensional functions and the details about these functions are presented in Table I.

TABLE I
Benchmark function details

No	Function	Optimal
	Unimodal functions	
1	Sphere function	-1400
2	Rotated high conditioned elliptic function	-1300
3	Rotated bent cigar function	-1200
4	Rotated discus function	-1100
5	Different powers function	-1000
	Basic multimodal functions	
6	Rotated Rosenbrock's function	-900
7	Rotated Schaffers F7 function	-800
8	Rotated Ackley's function	-700
9	Rotated Weierstrass function	-600
10	Rotated Griewank's function	-500
11	Rastrigin's function	-400
12	Rotated Rastrigin's function	-300
13	Non-Continuous rotated Rastrigin's function	-200
14	Schwefel's Function	-100
15	Rotated Schwefel's Function	100

EHO parameters were set as follows. Population size was 50 and the number of clans was 5. Parameter α was set to 0.5 and parameter β was 0.1. Maximal generation number was 5000. For each test, algorithms were executed 30 times and median, standard deviation, the best and the worst solutions were calculated, same as in [26].

We compared our proposed chaotic based methods with the particle swarm optimization [26] and the original EHO. In this paper, we introduced two chaotic maps to the EHO algorithm, circle map to implement CEHO-C and sinusoidal map SEHO-C. The obtained results are presented in Table II. Best results are in bold.

TABLE II
Comparison of PSO, EHO, CEHO-C and CEHO-S

Fun.		PSO	EHO	CEHO-C	CEHO-S
f_1	median	-1.400E+03	-1.400E+03	-1.400E+03	-1.400E+03
	std	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	best	-1.400E+03	-1.400E+03	-1.400E+03	-1.400E+03
	worst	-1.400E+03	-1.400E+03	-1.400E+03	-1.400E+03
f_2	median	3.500E+04	2.934E+04	8.665E+04	**1.470E+04**
	std	7.360E+04	8.328E+04	**2.397E+05**	2.572E+04
	best	7.597E+02	**1.853E+02**	6.063E+03	2.963E+03
	worst	4.755E+05	4.129E+05	**1.220E+05**	6.751E+04
f_3	median	2.670E+05	**1.284E+05**	1.337E+05	7.762E+04
	std	1.660E+07	6.834E+06	1.514E+07	**6.967E+05**
	best	-1.200E+03	**-1.158E+03**	3.844E+04	5.471E+03
	worst	8.251E+07	**1.795E+08**	3.505E+07	1.611E+06
f_4	median	7.769E+03	2.359E+03	6.118E+03	**-9.953E+02**
	std	4.556E+03	1.631E+03	1.416E+04	**1.818E+02**
	best	2.454E+02	1.195E+02	-9.738E+02	**-1.099E+03**
	worst	1.856E+04	5.270E+03	3.274E+04	**-6.570E+02**
f_5	median	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03
	std	3.142E-05	0.000E+00	2.778E+01	**4.957E-05**
	best	-1.000E+03	-1.000E+03	-1.000E+03	-1.000E+03
	worst	-1.000E+03	-1.000E+03	-9.433E+02	-1.000E+03
f_6	median	-8.902E+02	-8.256E+02	-8.995E+02	**-8.973E+02**
	std	4.974E+00	3.638E+00	7.499E-01	**7.893E-02**
	best	-9.000E+02	**-9.000E+02**	-8.974E+02	-8.996E+02
	worst	-8.898E+02	-8.898E+02	-8.935E+02	**-8.994E+02**
f_7	median	-7.789E+02	-7.582E+02	-7.640E+02	**-7.885E+02**
	std	1.327E+01	1.170E+01	1.976E+01	**4.589E+00**
	best	-7.974E+02	-7.697E+02	-7.723E+02	**-7.987E+02**
	worst	-7.434E+02	-7.382E+02	-7.232E+02	**-7.831E+02**
f_8	median	-6.789E+02	-6.797E+02	**-6.800E+02**	-6.798E+02
	std	6.722E-02	4.338E-03	**2.753E-03**	5.318E-02
	best	-6.789E+02	-6.797E+02	**-6.800E+02**	-6.799E+02
	worst	-6.796E+02	-6.797E+02	**-6.800E+02**	-6.797E+02
f_9	median	-5.952E+02	-5.969E+02	**-5.979E+02**	-5.923E+02
	std	1.499E+00	1.039E+00	6.312E-01	**7.699E-01**
	best	-5.987E+02	-5.991E+02	**-5.982E+02**	-5.926E+02
	worst	-5.929E+02	-5.929E+02	**-5.931E+02**	-5.910E+02
f_{10}	median	-4.999E+02	**-4.999E+02**	**-4.999E+02**	**-4.999E+02**
	std	2.713E-01	1.449E-01	**2.331E-02**	1.795E-02
	best	-4.999E+02	**-5.000E+02**	**-5.000E+02**	-4.999E+02
	worst	-4.989E+02	-4.984E+02	**-4.999E+02**	-4.998E+02
f_{11}	median	-3.891E+02	-3.907E+02	**-3.913E+02**	-3.473E+02
	std	5.658E+00	4.198E+00	**2.119E+00**	1.777E+01
	best	-3.970E+02	**-3.972E+02**	-3.870E+02	-3.781E+02
	worst	-3.731E+02	**-3.781E+02**	-3.740E+02	-3.353E+02
f_{12}	median	-2.861E+02	-2.870E+02	**-2.920E+02**	-2.682E+02
	std	6.560E+00	6.019E+00	**1.015E+00**	1.054E+01
	best	-2.970E+02	**-2.971E+02**	-2.970E+02	-2.731E+02
	worst	-2.682E+02	-2.623E+02	**-2.901E+02**	-2.483E+02
f_{13}	median	-1.792E+02	-1.801E+02	**-1.834E+02**	-1.398E+02
	std	9.822E+00	8.992E+00	**5.168E+00**	9.588E+00
	best	-1.946E+02	**-1.992E+02**	-1.952E+02	-1.434E+02
	worst	-1.523E+022	-1.617E+02	**-1.802E+02**	-1.201E+02
f_{14}	median	7.338E+02	2.914E+02	**2.758E+02**	1.093E+03
	std	1.282E+02	1.282E+02	**9.365E+01**	1.444E+02
	best	2.228E+02	-1.419E+02	**1.677E+02**	9.702E+02
	worst	1.109E+03	4.990E+02	**4.526E+02**	1.303E+03
f_{15}	median	8.743E+02	5.695E+02	**4.309E+02**	9.160E+02
	std	2.507E+02	2.429E+02	2.529E+02	**2.323E+02**
	best	4.372E+02	4.271E+02	**3.597E+02**	1.136E+03
	worst	1.705E+03	1.044E+03	**7.240E+02**	1.346E+03

As it can be seen from Table II, all algorithms found the optimal function value for f_1 (sphere). Standard deviation was 0 which means that the optimal function value was determined in each run. EHO algorithm found exact optimal value for f_5 with standard deviation 0. EHO, chaotic based EHO algorithms, as well as the PSO, were not able to find solutions for test functions f_2 and f_3. For function f_4 EHO with sinusoidal map achieved the best results. Similar results were obtained for test function f_7.

For functions f_6 and f_8 PSO and EHO achieved the same median and the best solution. For functions f_8 to f_15 EHO algorithm with circle maps obtained the best results compared to the other methods.

IV. Conclusion

In this paper we proposed chaos based elephant herding optimization algorithm for solving unconstrained global optimization problems. We introduced two different one-dimensional chaotic maps, circle and sinusoidal maps, to the original EHO algorithm. The proposed algorithms were tested and compared on 15 standard benchmark functions (CEC 2013). Based on the results, chaotic based EHO outperforms particle swarm optimization and the original EHO in almost all cases. For functions f_1 to f_7 circle based EHO algorithm performed better then other algorithms, while for benchmark functions f_8 to f_15 the sinusoidal EHO outperformed PSO, EHO and circle based EHO. In further work, different chaotic maps and their combinations can be used and compared to other chaotic optimization algorithms.

Acknowledgment

This research is supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, Grant No. III-44006.

References

[1] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications, LNCS*. Springer, Heidelberg, 2009, vol. 5792, pp. 169–178.

[2] M. Tuba and N. Bacanin, "Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems," *Neurocomputing*, vol. 143, pp. 197–207, 2014.

[3] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *ICSI 2010, LNCS*, Y. Tan, Y. Shi, and K. C. Tan, Eds., vol. 6145. Springer, Heidelberg, 2010, pp. 355–364.

[4] M. Tuba, N. Bacanin, and A. Alihodzic, "Multilevel image thresholding by fireworks algorithm," in *25th international conference Radioelektronika*. IEEE, 2015, pp. 326–330.

[5] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report - TR06*, pp. 1–10, 2005.

[6] M. Tuba and R. Jovanovic, "Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem," *International Journal of Computers, Communications & Control*, vol. 8, no. 3, pp. 477–485, 2013.

[7] E. Tuba, M. Tuba, and M. Beko, "Node localization in ad hoc wireless sensor networks using fireworks algorithm," in *5th International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 2016, pp. 223–229.

[8] E. Tuba, L. Mrkela, and M. Tuba, "Support vector machine parameter tuning using firefly algorithm," in *26th International Conference Radioelektronika*. IEEE, 2016, pp. 413–418.

[9] 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.

[10] E. Tuba, M. Tuba, and M. Beko, "Support vector machine parameters optimization by enhanced fireworks algorithm," in *ICSI 2016, LNCS*, Y. Tan, Y. Shi, and B. Niu, Eds. Springer, Cham, 2016, vol. 9712, pp. 526–534.

[11] E. Tuba, M. Tuba, and E. Dolicanin, "Adjusted fireworks algorithm applied to retinal image registration," *Studies in Informatics and Control*, vol. 26, no. 1, pp. 33–42, 2017.

[12] N. Bacanin and M. Tuba, "Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint," *The Scientific World Journal, special issue Computational Intelligence and Metaheuristic Algorithms with Applications,*, vol. 2014, no. Article ID 721521, p. 16 pages, 2014.

[13] A. Gandomi, X.-S. Yang, S. Talatahari, and A. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 89–98, 2013.

[14] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.

[15] M. Mitic, N. Vukovic, M. Petrovic, and Z. Miljkovic, "Chaotic fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 89, pp. 446–458, 2015.

[16] C. Gong, "Chaotic adaptive fireworks algorithm," in *ICSI 2016, LNCS*, Y. Tan, Y. Shi, and B. Niu, Eds. Springer, Cham, 2016, vol. 9712, pp. 515–525.

[17] D. Tian, "Particle swarm optimization with chaos-based initialization for numerical optimization," *Intelligent Automation & Soft Computing*, pp. 1–12, 2017.

[18] L. Huang, S. Ding, S. Yu, J. Wang, and K. Lu, "Chaos-enhanced cuckoo search optimization algorithms for global optimization," *Applied Mathematical Modelling*, vol. 40, no. 5, pp. 3860–3875, 2016.

[19] X. Yuan, J. Zhao, Y. Yang, and Y. Wang, "Hybrid parallel chaos optimization algorithm with harmony search algorithm," *Applied Soft Computing*, vol. 17, pp. 12–22, 2014.

[20] Q. Wu, "A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM," *Expert Systems with Applications*, vol. 38, no. 1, pp. 184–192, 2011.

[21] F. Liu and Z. Zhou, "A new data classification method based on chaotic particle swarm optimization and least square-support vector machine," *Chemometrics and Intelligent Laboratory Systems*, vol. 147, pp. 147–156, 2015.

[22] G.-G. Wang, S. Deb, X.-Z. Gao, and L. D. S. Coelho, "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour," *International Journal of Bio-Inspired Computation*, vol. 8, no. 6, pp. 394–409, 2016.

[23] E. Tuba, A. Alihodzic, and M. Tuba, "Multilevel image thresholding using elephant herding optimization algorithm," in *14th International Conference on Engineering of Modern Electric Systems (EMES)*. IEEE, 2017, pp. 240–243.

[24] E. Tuba and Z. Stanimirovic, "Elephant herding optimization algorithm for support vector machine parameters tuning," in *IEEE 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2017, pp. 1–4.

[25] M. A. Sarwar, B. Amin, N. Ayub, S. H. Faraz, S. U. R. Khan, and N. Javaid, "Scheduling of appliances in home energy management system using elephant herding optimization and enhanced differential evolution," in *International Conference on Intelligent Networking and Collaborative Systems*. Springer, 2017, pp. 132–142.

[26] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2013, pp. 2337–2344.