Hybridized Monarch Butterfly Algorithm for Global Optimization Problems

IVANA STRUMBERGER
Singidunum University
Faculty of Informatics and Computing
Danijelova 32, 11000 Belgrade
SERBIA
istrumberger@singidunum.ac.rs

MARKO SARAC
Singidunum University
Faculty of Informatics and Computing
Danijelova 32, 11000 Belgrade
SERBIA
msarac@singidunum.ac.rs

DUSAN MARKOVIC
Singidunum University
Faculty of Informatics and Computing
Danijelova 32, 11000 Belgrade
SERBIA
dmarkovic@singidunum.ac.rs

NEBOJSA BACANIN
Singidunum University
Faculty of Informatics and Computing
Danijelova 32, 11000 Belgrade
SERBIA
nbacanin@singidunum.ac.rs

Abstract: This paper introduces hybridized monarch butterfly optimization algorithm for solving global optimization problems. Despite of the fact that the monarch butterfly optimization algorithm is relatively new approach, it has already showed great potential when tackling NP-hard optimization tasks. However, by analyzing original monarch butterfly algorithm, we noticed some deficiencies in the butterfly adjusting operator that in early iterations exceedingly directs the search process towards the current best solution. To overcome this deficiency, we incorporated firefly's algorithm search mechanism into the original monarch optimization approach. We tested our algorithm on six standard global optimization benchamarks, and performed comparative analysis with original monarch butterfly optimization, as well as with other five state-of-the-art metaheuristics. Experimental results are promising.

Key–Words: monarch butterfly optimization, algorithms, global optimization, swarm intelligence, metaheuristics

1 Introduction

Since almost every practical task can be modeled as an optimization problem, in the last few decades, numerical optimization is in the main focus of many researcher around the world. Also, many techniques, methods and algorithms were develop for tackling numerical problems.

In general, numerical optimization problems can be divided into two main categories. Continuous (global) tasks represent first class of numerical problems, and in this case variables take real values values. Contrarily, problems which variables can take only discrete (integer) values are known in the literature as combinatorial tasks.

Continuous (global) problems, when taking into account possible constraints, can further be divided into unconstrained and constrained. Unconstrained (or bound constrained) optimization can be defined as D-dimensional minimization or maximization problem which can be expressed as:

$$\min(\max) f(x),\ x = (x_1, x_2, x_3, ..., x_D) \in S, \quad (1)$$

where x is denotes real vector with $D \geq 1$ components and $S \in R^D$ is an D-dimensional hyper-rectangular search space constrained by lower and upper bounds:

$$lb_i \leq x_i \leq ub_i,\ \ i \in [1, D] \quad (2)$$

In Eq. (2), lb_i and ub_i represent lower and upper bounds of i-th vector component, respectively.

Many numerical optimization problems belong to the group of NP-hard (nondeterministic polynomial time) optimization. A problem is NP-hard if an algorithm for its solving can be translated into one for solving any NP-problem. Thus, NP-hard denotes "at least as hard as any NP-problem", while in fact it might be harder. Some of the well-known combinatorial NP-hard problems are subset sum problem and traveling salesman problem (TSP).

NP-hard problems can not be solved in a satisfying amount of computational time by employing standard, deterministic algorithms (algorithms that for the same input always generate the same output). For tackling NP-hard problems, many metaheuristics approaches were developed. Metaheuristics algorithms can find suboptimal (satisfying) solutions in an acceptable time period, but can not guarantee that the optimum solution will be found.

1.1 Swarm intelligence

Many metaheuristics can be found in the literature, and one of the most promising group of metaheuristics is swarm intelligence. Swarm intelligence is branch of artificial intelligence, that simulate the group of organisms from the nature, relying on four self-organization principles: positive feedback, negative feedback, multiple interactions and fluctuations [1]. Swarm intelligence are population-based, iterative and stochastic optimization methods. During last few decades, many swarm intelligence algorithms were developed mimicking flock of birds and fish, groups of cuckoo birds, bats and moths, colonies of bees and ant, herds of elephants, etc.

Well-known representative of swarm intelligence is artificial bee colony (ABC) that simulates the behavior of hives of honey bee swarms [2]. In the ABC approach, three types of bees conduct the search process: employees, onlookers and scouts. It was applied to numerous real world problems such as RFID network planning [3], [4], constrained optimization problems [5].

Firstly proposed by Yang in 2009 [6], the firefly algorithm (FA) emulates lighting behavior of firefly insects, when less brighter firefly moves to the direction of firefly with greater brightness. The FA showed outstanding performance on many benchmark and real-world problems [7], [8], [9], [10]. Proposed by the same author as the FA, cuckoo search (CS) algorithm was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds [11]. This well-known approach has also many applications for different kinds of problems [12], [13].

Herding behavior of elephants was the main source of inspiration for devising elephant herding optimization (EHO) algorithm [14]. As relatively new swarm intelligence metaheuristics, the EHO proved to be robust optimization method for various benchmark and real-life problems. EHO was also applied on support vector machine parameters tuning [15], [16], multilevel image thresholding [17], path planning problem [18] and static drone placement [19].

Besides above mentioned, according to the literature survey, there are also many other swarm intelligence algorithms that were successfully applied to various tasks like general benchmark problems portfolio optimization [20], [21], node localization in wireless sensor networks [22], image processing [23], path planning [24], machine learning optimization [25], RFID network planning [3], etc.

In this paper, we propose hybridized monarch butterfly optimization (MBO) algorithm for global optimization problems. The MBO is novel swarm intelligence metaheuristic, firstly proposed by Wang and Deb in 2015 [26]. The MBO emulates the migration behavior of monarch butterflies.

During empirical tests with the original MBO, we noticed some deficiencies in the search process, and to overcome these deficiencies, we hybridized original MBO with the FA metaheuristics.

The rest of the paper has following structure: after Introduction, we present original and hybridized MBO in Section 2 and 3, respectively, experimental results and comparative analysis with other approaches are given in Section 4, while Section 5 concludes this paper.

2 Original monarch butterfly optimization algorithm

Monarch butterflies inhabit areas in the territory of North America and belong to the family of Nymphalidae. The butterflies are one of the most beautiful butterflies in the world and can be easily distinguished by orange and black patterns.

During each year, monarch butterflies migrate twice. First migration starts every year in August, when monarch butterflies fly thousand of miles from the USA and southern part of Canada to Mexico. Second migration takes place in the spring, when butterflies migrates from Mexico to USA and Canada.

The MBO's search process models two properties of monarch butterflies: Lévy flights, and the process of laying eggs by female butterflies for generating offspring [27]. Lévy flights are performed by some (not all) butterflies during the migration period.

MBO was firstly proposed in 2015 by Wang and Deb [26] for solving global optimization problems. The very first implementation of MBO was tested on thirty-eight global optimization benchmarks, where MBO outperformed other state-of-the-art algorithms and proved to be robust and efficient optimization method.

In most cases, nature-inspired algorithms, that model behavior and characteristics of natural systems, employ simplifications by using idealized rules. The MBO algorithm apply four rules that simplify migration behavior of monarch butterflies [26]. The search process of the MBO metaheuristics is guided by two operators: migration operator and butterfly adjusting operator.

For the purpose of migration operator's implementation, the migration process of monarch butterflies from the nature is idealized as follows: the whole population of monarch butterflies stay in Land 1 for 5 months (from April to August), and stay in Land 2 for 7 months (from September to March).

In order to be consistent with swarm intelligence terminology, terms Land 1 and Land 2 are substituted with subpopulation 1 and subpopulation 2, respectively.

The number of monarch butterflies in subpopulation 1 and subpopulation 2 are calculated by using Eq. (3) and Eq. (4), respectively [26]:

$$ceil(p \cdot NP) \cdot NP_1 \tag{3}$$

$$(NP - NP_1) \cdot NP_2, \tag{4}$$

where function $ceil(a)$ rounds argument a to the nearest integer greater than or equal to a, NP denotes the total size of the population (number of butterflies), NP_1 and NP_2 are number of monarch butterflies in subpopulation 1 and subpopulation 2, respectively, and p represents the ratio of monarch butterflies in the subpopulation 1.

The MBA's migration can be mathematically expressed as [26]:

$$x_{i,k}^{t+1} = x_{r_1,k}^{t}, \tag{5}$$

where $x_{i,k}^{t+1}$ is the k-th element (component) of the x_i individual at generation $t+1$, and $x_{r_1,k}^{t}$ is the k-th parameter of the individual x_{r_1} at current generation t. Parameter $r1$ represents individual that is randomly selected from subpopulation 1.

When the expression $r \leq p$ is satisfied, the k-th parameter of the newly created butterfly is calculated by employing Eq. (5).

The ratio r can be calculated as [26]:

$$r = rand \cdot peri, \tag{6}$$

where $peri$ denotes the migration period, and $rand$ is uniformly distributed pseudo-random number.

On contrarily, if the expression $r \geq p$ holds, the parameter k of the newly generated monarch butterfly is calculated as [26]:

$$x_{i,k}^{t+1} = x_{r_2,k}^{t}, \tag{7}$$

where $x_{r_2,k}^{t}$ represents the k-the parameter of the randomly chosen butterfly r_1 from subpopulation 2 in the current generation t.

The second mechanism that guides monarch butterfly's search process is butterfly adjusting operator. This operator operates as follows: for all parameters in individual (butterfly) j, if the pseudo-random number $rand$ is smaller than or equal to p, the new solution is generated using the following equation [26]:

$$x_{j,k}^{t+1} = x_{best,k}^{t}, \tag{8}$$

where $x_{j,k}^{t+1}$ is the k-th parameter of the new solution j, and $x_{best,k}^{t}$ is the k-th parameter of current best solution in the whole population.

In the second case, if the uniformly distributed number $rand$ is greater than p, the new solution is by using the following expression [26]:

$$x_{j,k}^{t+1} = x_{r_3,k}^{t}, \tag{9}$$

where $x_{r_3,k}^{t}$ denotes the k-th parameter of randomly selected solution r_3 from Subpopulation 2, and $r_3 \in \{0, 1, 2, ..., NP_2\}$.

Finally, if the condition $rand \geq BAR$ holds, the k-th parameter of the child solution is created as [26]:

$$x_{j,k}^{t+1} = x_{j,k}^{t} + \alpha \times (dx_k - 0.5), \tag{10}$$

where BAR represents butterfly adjusting rate, and dx is the walk step of the monarch butterfly j that can be calculated using Lévy flights [26].

3 Hybridized monarch butterfly optimization algorithm

By conducting empirical tests, and performing theoretical analysis of MBO's behavior, we noticed deficiency in the search process conducted by butterfly adjusting operator. In early stages of algorithm's execution, the search process that is exceedingly directed towards the current best solution in the population (Eq. 8), in some of the algorithm's run generate poor results.

In this case, when algorithm by accident, in early iterations, hits the right part of the search space, this search process around the current best yields good performance. Unfortunately, in most algorithm's runs, the MBO does not hit the right part of the search space in early iterations, which leads to the worse mean values with high dispersion.

To tackle with this deficiency, in early iterations of algorithm's execution, we incorporated FA's search equation in the original MBO, that replaces search process conducted by using Eq. (8). In this way, we modified butterfly adjusting operator.

In the FA metaheuristics, the movement of a firefly i (process of exploration and exploitation) towards the brighter, and thus more attractive firefly j, for each solution's parameter, is determined by [28]:

$$x_i^{t+1} = x_i^t + \beta_0 r^{-\gamma r_{i,j}^2}(x_j^t - x_i^t) + \alpha(rand - 0.5), \quad (11)$$

where $t+1$ is the next iterations, t is the current iteration, β_0 is attractiveness at $r=0$, α is randomization parameter, $rand$ is random number uniformly distributed between 0 and 1, and $r_{i,j}$ is distance between fireflies i and j.

By incorporating FA's search equation (Eq. (11)) into the MBO, we developed hybridized approach named MBO firefly search (MBO-FS). Pseudo-code of the MBO-FS metaheuristics is shown in Algorithm 1

Algorithm 1 Pseudo-code of the MBO-FS algorithm

Initialization. Set the iterations counter $t = 1$; generate the population P of NP monarch butterfly individuals randomly; set the maximum generation number $MaxGen$, monarch butterfly number in land 1 NP_1 and in land 2 $NP2$, max step length S_{max}, butterfly adjusting rate BAR, migration period $peri$ and the migration ratio p.
Fitness evaluation. Evaluate each monarch butterfly against the objective function and calculate fitness.
while $t < MaxGen$ **do**
  Sort all individuals in the population according to its fitness.
  Divide the whole population into subpopulation 1 and subpopulation 2.
  for $i = 1$ **to** NP_1 (all butterflies in the subpopulation 1) **do**
    Generate new individuals in subpopulation 1 by using butterfly migration operator
  end for
  for $j = 1$ **to** NP_2 (all butterflies in the subpopulation 2) **do**
    if $t < MaxGen \cdot 0.5$ **then**
      Generate new individuals in subpopulation 2 by using modified butterfly adjusting operator
    else
      Generate new individuals in subpopulation 2 by using butterfly adjusting operator
    end if
  end for
  Merge newly generated subpopulation 1 and subpopulation 2 into one whole population.
  Evaluate the population according to the newly updated positions.
  Increase the iteration counter t by one.
end while
return Best individual in the whole population

4 Experimental results and comparative analysis

MBO-FS parameters were set as follows: $S_{max} = 1.0$, butterfly adjusting rate (BAR) is set to 5/12, migration period ($peri$) is adjusted to 1.2, and the migration ratio r is set to 5/12. Maximum number of generations (iterations) in one algorithm's execution ($MaxGen$) is set to 200. Population size parameters were set as: total population number NP=50, subpopulation 1 NP_1=21 and subpopulation 2 NP_2=29. With 200 generations and 50 individuals, total number of function evaluations is 10.000.

We conducted experiments on six standard unconstrained benchmark functions: $Ackley$ (f0), $Dixon\&Price$ (f1), $Fletcher-Powell$ (f2), $Griewank$ (f3), $Perm$ (f4) and $Step$ (f5). Comparative analysis was performed with the original MBO algorithm, as well as with ABC, biogeography-based optimization (BBO) differential evolution (DE), particle swarm optimization (PSO) and stud genetic algorithms (SGA).

The algorithm was executed in 30 independent runs, and we measured best, mean and worst results' values. Comparative analysis of best, mean and worst obtained results are given in Tables 1, 2 and 3, respectively.

Table 1: Comparison of best values

F.	ABC	BBO	DE	PSO	SGA	MBO	MBO-FS
f0	13.35	2.51	16.48	17.05	2.51	2.1E-8	**0.3E-8**
f1	14.0E6	4.6E3	1.2E6	4.1E6	2.1E3	**0.67**	0.89
f2	1.2E5	3.7E4	1.7E5	3.3E5	3.9E4	3.4E4	**6.13E3**
f3	30.93	1.79	10.96	34.86	1.37	**1.00**	1.19
f4	1.4E45	6.0E51	3.7E37	3.7E43	6.0E51	3.0E32	**9.6E31**
f5	16.00	1.00	6.00	20.00	1.00	1.00	1.00

Table 2: Comparison of mean values

F.	ABC	BBO	DE	PSO	SGA	MBO	MBO-FS
f0	16.45	3.77	18.26	18.44	4.33	2.4E6	**3.1E-7**
f1	4.6E7	7.7E4	3.8E6	1.4E7	1.1E4	0.67	**0.63**
f2	2.7E5	**7.0E4**	2.5E5	5.0E5	8.3E4	1.6E5	9.3E4
f3	85.88	3.33	21.42	73.02	2.19	**1.00**	1.15
f4	1.2E51	6.1E51	4.5E45	4.5E47	6.0E51	2.5E37	**2.5E35**
f5	35.68	1.16	9.26	27.50	1.44	1.00	1.00

According to presented empirical resuts, our hybridized MBO-FS in average outperforms all other approaches including the original MBO. Only in some cases, for example worst and best results for f1

Table 3: Comparison of worst values

F.	ABC	BBO	DE	PSO	SGA	MBO	MBO-FS
f0	17.85	5.77	18.97	18.88	6.42	1.3E5	**0.95E-6**
f1	1.0E8	2.8E5	9.0E6	3.2E7	4.0E4	**0.67**	0.69
f2	4.0E5	1.2E5	3.3E5	8.2E5	1.8E5	**3.9E5**	4.3E5
f3	136.66	5.83	31.13	104.06	3.98	1.00	**0.96**
f4	6.0E51	1.0E52	6.0E46	3.1E48	6.0E51	3.5E38	**3.4E37**
f5	49.00	2.00	14.00	36.00	4.00	1.00	1.00

($Dixon\&Price$) benchmark, original MBO performs better than hybridized MBO-FS.

As a conclusion, we state that, our proposed hybridization enhanced basic MBO algorithm by introducing FA's search equation. In this way, the balance between exploitation and exploration is better adjusted, especially in early stages of algorithm's execution.

5 Conclusion

In this paper, we introduced hybridized monarch butterfly optimization algorithm for solving global optimization problems. Despite of the fact that the monarch butterfly optimization algorithm is relatively new approach, it has already showed great potential when tackling NP-hard optimization tasks. However, by analyzing original monarch butterfly algorithm, we noticed some deficiencies in the butterfly adjusting operator that in early iterations exceedingly directs the search process towards the current best solution. To overcome this deficiency, we incorporated firefly's algorithm search mechanism into the original monarch optimization approach. We named our approach MBO firefly search (MBO-FS).

We tested MBO-FS algorithm on six standard global optimization benchamarks, and performed comparative analysis with original monarch butterfly optimization, as well as with other five state-of-the-art metaheuristics. In average, MBO-FS outperformed all other metaheuristics included in comparative analysis, and as the final conclusion, we state that the MBO-FS has great potential in dealing with NP-hard tasks.

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

References:

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, 1999.

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

[3] M. Tuba and N. Bacanin, "Hybridized bat algorithm for multi-objective radio frequency identification (RFID) network planning," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC 2015)*, May 2015.

[4] N. Bacanin and M. Tuba, "Multiobjective RFID network planning by artificial bee colony algorithm with genetic operators," *Advances in Swarm and Computational Intelligence, ICSI 2015, Lecture Notes in Computer Science*, vol. 9140, pp. 247–254, 2015.

[5] I. Brajevic, M. Tuba, and M. Subotic, "Improved artificial bee colony algorithm for constrained problems," in *Proceedings of the 11th WSEAS International Conference on Evolutionary Computing*, pp. 185–190, 2010.

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

[7] I. Strumberger, N. Bacanin, and M. Tuba, "Enhanced firefly algorithm for constrained numerical optimization, ieee congress on evolutionary computation," in *Proceedings of the IEEE International Congress on Evolutionary Computation (CEC 2017)*, pp. 2120–2127, June 2017.

[8] E. Tuba, M. Tuba, and M. Beko, "Mobile wireless sensor networks coverage maximization by firefly algorithm," in *27th International Conference Radioelektronika*, pp. 1–5, IEEE, 2017.

[9] M. Tuba and N. Bacanin, "JPEG quantization tables selection by the firefly algorithm," in *International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 153–158, IEEE, 2014.

[10] E. Tuba, M. Tuba, and M. Beko, "Two stage wireless sensor node localization using firefly algorithm," in *Smart Trends in Systems, Security and Sustainability, LNNS*, vol. 18, pp. 113–120, Springer, 2018.

[11] X.-S. Yang and S. Deb, "Cuckoo search via levy flights," in *Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210–214, 2009.

[12] I. Brajevic and M. Tuba, "Cuckoo search and firefly algorithm applied to multilevel image thresholding," in *Cuckoo Search and Firefly Algorithm: Theory and Applications* (X.-S. Yang, ed.), vol. 516 of *Studies in Computational Intelligence*, pp. 115–139, Springer International Publishing, 2014.

[13] N. Bacanin, "Implementation and performance of an object-oriented software system for cuckoo search algorithm," *International Journal of Mathematics and Computers in Simulation*, vol. 6, pp. 185–193, December 2010.

[14] G.-G. Wang, S. Deb, and L. dos S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 1–5, December 2015.

[15] E. Tuba and Z. Stanimirovic, "Elephant herding optimization algorithm for support vector machine parameters tuning," in *Proceedings of the 2017 International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–5, June 2017.

[16] E. Tuba, I. Ribic, R. Capor-Hrosik, and M. Tuba, "Support vector machine optimized by elephant herding algorithm for erythemato-squamous diseases detection," *Procedia Computer Science*, vol. 122, pp. 916–923, 2017.

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

[18] A. Alihodzic, E. Tuba, R. Capor-Hrosik, E. Dolicanin, and M. Tuba, "Unmanned aerial vehicle path planning problem by adjusted elephant herding optimization," in *25th Telecommunication Forum (TELFOR)*, pp. 1–4, IEEE, 2017.

[19] I. Strumberger, N. Bacanin, M. Beko, S. Tomic, and M. Tuba, "Static drone placement by elephant herding optimization algorithm," in *Proceedings of the 24th Telecommunications Forum (TELFOR)*, November 2017.

[20] N. Bacanin and M. Tuba, "Fireworks algorithm applied to
constrained portfolio optimization problem," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC 2015)*, May 2015.

[21] I. Strumberger, N. Bacanin, and M. Tuba, "Constrained portfolio optimization by hybridized bat algorithm," in *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pp. 83–88, Jan 2016.

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

[23] E. Tuba, M. Tuba, and E. Dolicanin, "Adjusted fireworks algorithm applied to retinal image registration," *Studies in Informatics and Control*, vol. 26, pp. 33–42, March 2017.

[24] E. Dolicanin, I. Fetahovic, E. Tuba, R. Capor-Hrosik, and M. Tuba, "Unmanned combat aerial vehicle path planning by brain storm optimization algorithm," *Studies in Informatics and Control*, vol. 27, no. 1, pp. 15–24, 2018.

[25] E. Tuba, M. Tuba, and D. Simian, "Adjusted bat algorithm for tuning of support vector machine parameters," in *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2225–2232, IEEE, 2016.

[26] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, pp. 1–20, May 2015.

[27] G. A. Breed, P. M. Severns, and A. M. Edwards, "Apparent power-law distributions in animal movements can arise from intraspecific interactions," *Journal of the The Royal Society Interface*, vol. 12, February 2015.

[28] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.