Fifth International Conference
Modelling and Development of Intelligent Systems
June 23 - 25, 2017
"Lucian Blaga" University
Sibiu - Romania

Moth Search Algorithm for Bound Constrained Optimization Problems

Eva Tuba, Romana Capor-Hrosik, Adis Alihodzic, Marko Beko, Raka Jovanovic

Abstract

Numerous real life problems represent hard optimization problems that cannot be solved by deterministic algorithms. In the past decades, various different methods were proposed for these kind of problems and one of the methods are nature inspired algorithms, especially swarm intelligence algorithms. Moth search optimization algorithm (MSO) is on of the recent swarm intelligence algorithm that has not been thoroughly researched. In this paper we tested MSO algorithm on 15 standard benchmark functions and compared results with particle swarm optimization algorithm. Comparison show that MSO has good characteristics and it outperformed other approach from literature.

1 Introduction

Optimization represents an important field in applied mathematics. Most of the real world problems can be described as an optimization problem where the goal is to find minimum or maximum of the objective (fitness) function. The goal is to find the solution $x* \in A$ for the objective function $F : A \rightarrow R$ where $A \subset R^N$ that satisfy the following condition:

$$K(x) \leq K(x*), \tag{1}$$

in the case of the function maximization or, in the case of minimization:

$$K(x) \geq K(x*). \tag{2}$$

For solving simple optimization problems, various deterministic models were developed. The problem is that real world optimization problems are usually rather difficult and these deterministic methods cannot find the optimal solution or at least, they cannot find one in reasonable time. One well known example of hard optimization problem is traveling salesman which can be solved by heuristic (checking all possible solutions). The complexity of deterministic method is N! which will make calculation time unreasonably long even for rather small problem dimension.

For hard optimization problem, so-called NP difficult, it is easy to test the solution, to find the value of the fitness function, but it is hard to find the optimal one. In order to find the solution for these problems, stochastic algorithms have to be used. These algorithms are using random elements in their execution along with the set of rules for finding solution inside the domain of possible solutions. Due the random factors, stochastic algorithms produce different solutions in each run thus they have to be allowed to run long enough, in order to give ”good enough” each time. Common practice is to run stochastic algorithm several times and as final solution to use the average of the obtained results.

One large group of stochastic algorithms are based on imitation of natural phenomena. Empirically, it was shown that in this way, good results can be obtained, but it is not always completely understood how. Nowadays, nature inspired algorithms are common research topic and numerous methods were developed. They can be roughly categorized into three groups: evolutionary, artificial immune systems and swarm intelligence.

Evolution algorithms have been inspired by the evolutionary process, i.e. on the idea of the survival of fittest agent. Each generation of the solutions is made by combining the best solutions from previous generation or by mutating them while the worst solutions are not selected to survive. Mutation represents the random factor in this algorithms. By numerous iterations of breeding, where each generation is closer to the optimal solution, it can be expected that after enough iterations, ”good enough” solution will be found. Well known algorithms in this group are genetic algorithm, differential evolution, genetic programming, etc.

Similar concept to the described one is used in artificial immune systems where negative selection is the main characteristic. These algorithms search for bad solutions and eliminate them from population which is the process inspired by natural immune systems in living beings.

Swarm intelligence algorithms are inspired by different phenomena that include numerous agents able to perform simple operations and communicate between them self. In these algorithms, movement of each organism is influenced by the information that it collected, by the information obtained from other agents and random factor. Algorithms can be inspired by animals in nature, music harmony, brain storm process, fireworks explosion, etc. Particle swarm optimization [1] and ant colony optimizations [2] are among the first swarm intelligence algorithms. Some other well known algorithms in this group are artificial bee colony algorithm, harmony search, bat algorithm, cuckoo search and many others.

In this paper we tested one of the recent swarm intelligence algorithm, moth search algorithm (MSA) proposed in 2016 by Gai Ge Wang. MSA is tested on benchmark functions CEC 2013 and compared to particle swarm optimization algorithm.

The rest of the paper is organized as follows. In Section 2, a short review of the swarm intelligence algorithms is given. Moth search algorithm is defined in Section 3. Experimental results are presented in Section 4 while the conclusion and future research are given in Section 5.

2 Swarm intelligence algorithms

Real world optimization problems are usually high dimensional and with rather difficult fitness function with large number of local extremes. In order to find the global optimum, optimization method needs to have a mechanize to escape from local optima. All optimization metaheuristics have two parts: exploration and exploitation, i.e. global and local search. Exploration ensures the search of the entire search space and it finds to promising areas. Around the found good solutions by the global search, i.e. promising areas, exploitation is used to try to find a better solutions.

Swarm algorithms use collaborative behavior of the simple individuals for solving the optimization problem. Swarms of ants and bees in their search for the path between home and food source, elephant population herding behavior and similar phenomena were the inspiration for these methods.

Artificial bee colony (ABC) imitates the foraging behavior of honey bee swarm. Exploitation and exploration processes were implemented by using three different types of bees: employed, onlookers and scouts. ABC algorithm have been shown to be rather effective for solving optimization problems [3]. Many upgraded and enhanced versions of ABC were proposed [4], [5] and also parallelized version in [6].

Cuckoo search (CS) represents swarm intelligence proposed by Yang and Deb in 2009 [7]. CS algorithm simulates the food search process by utilizing the Lévy flights. It was applied to numerous problems and it proved to be robust optimization method for the global optimization [8], image processing [9], engineering problems [10], etc.

Bat algorithm (BA) was also proposed by Yang in 2010 [11]. BA was inspired by the echolocation behavior of the bats. Algorithm parameters that controls exploration and exploitation are pulse rates of emission and loudness. Original and modified versions were used in various applications such as parameter tuning for support vector machine [12], handwritten digit recognition [13], etc.

Hybridization is rather common method for improving the quality of the swarm intelligence algorithm. For example, hybrid of ABC and firefly algorithm was proposed for portfolio optimization problem in [14] while in [15] it was used for problems with entropy constraint [16]. Seeker optimization algorithm (SOA) algorithm for global optimization was hybridized with ABC in [17] and in [18] hybrid between SOA and firefly algorithm was proposed.

3 Moth Search Optimization Algorithm

Moth search (MS) algorithm is recent member of the swarm intelligence algorithms. It was proposed by Wang in 2016 [19]. Inspiration for the MS algorithm was the phototaxis and Lévy flights of the moths. It was tested on numerous benchmark function such as IEEE CEC 2005 and IEEE CEC 2011 and compared to five metaheuristic algorithms. Based on the results presented in [19], it can be concluded that it has great potential for solving optimization problems. In this paper MS was applied to the newer benchmark function, IEEE CEC 2013, in order to test the quality of the algorithm for the more complex tasks.

As it was mentioned, MS algorithm was inspired by the moth behavior in the nature where phototaxis and Lévy flights represent the most significant characteristics of moths which was used for implementing an optimization algorithm.

Phototaxis represent the moths tendency to fly around the light source. They will fly in a straight line in order to stay at a fixed angle to the celestial light [19]. Lévy flight is one of the most important flight patterns in natural surroundings. The form of Lévy flights can be approximated by the power law distribution over a range of scales with the feature of exponents close to 3/2. Lévy flights was introduced to other optimization algorithms such as cuckoo search [7], firefly algorithm [20], krill herd optimization algorithm [21], bat algorithm [22], etc.

Lévy flights represent type of random walk where the step length is drawn from Lévy distribution modeled in the form of a power-law formula [19]:

$$L(s) \sim |s|^{-\beta}, \tag{3}$$

where $\beta \in [0, 3]$ is an index.

In [23], movements with $\beta \approx 1.5$ for Lévy flights were used which was also accepted in this paper.

Phototaxis and Lévy flights from moths in nature were used for implementing exploitation and exploration.

The best solution in the population is considered as the light source. The solutions (moths) closer to the best one will fly around the best moth it in the form of Lévy flights. New solutions, i.e. movements of the previous, are defined by the following equation [19]:

$$x_i^{t+1} = x_i^t + \alpha L(s), \tag{4}$$

where x_i^{t+1} and x_i^t are i^{th} solutions in the generation $t + 1$ and t, respectively. Step size based on the Lévy distribution is $L(s)$. Parameter α represents the scale factor and its value depends on the considered optimization problem. In [19], α was determined by:

$$\alpha = S_{max}/t^2, \tag{5}$$

where S_{max} represents the maximum walk step.

Lévy distribution in Eq. (4) is calculated by the following equation [19]:

$$L(s) = \frac{(\beta - 1)\Gamma(\beta - 1)\sin(\frac{\pi(\beta-1)}{2})}{\pi s^{\beta}}, \tag{6}$$

where Γ stands for the gamma function and $s > 0$.

Solutions that are on large distance from the best solution in the current generation fly towards the light source in line. These solutions are generated by the following formula [19]:

$$x_i^{t+1} = \lambda \times (x_i^t + \phi \times (x_{best}^t - x_i^t)), \tag{7}$$

where x_{best}^t is the best solution in generation t while ϕ and λ represent acceleration and scale factors, respectively.

Some solutions are defined by moving them beyond the best solutions in the current generation. These new solutions are generated by the following equation [19]:

$$x_i^{t+1} = \lambda \times (x_i^t + \frac{1}{\phi} \times (x_{best}^t - x_i^t)) \tag{8}$$

Initial population is randomly generated. In each iteration, new set of solutions is calculated by updating the previous solution by using Eq. (4), Eq. (7) or Eq. (8). In this paper we used the same simplified process as in the original paper [19]. Whole population was divided into two equal groups based on their fitness function values. In the first group, new solutions are obtained by the Lévy flights (Eq. (4)). The second group where are the solutions with worse fitness function values, update their solutions b Eq. (7) or Eq. (8) with possibility of $p = 0.5$ [19].

4 Experimental Results

Moth search algorithm was implemented by using Matlab R2016a and experiments were conducted on the platform with Intel ® Core^{TM} i7-3770K CPU at 4GHz, 8GB RAM, Windows 10 Professional OS.

MS algorithm was tested on fifteen standard benchmark functions proposed for CEC 2013 competition [24]. Used function along with their optimal fitness function values are listed in Table 1. Five unimodal and ten basic modal functions were used.

MS algorithm (MSA) was compared to the results obtained by particle swarm optimization (PSO) reported in [25]. For each test function algorithm was run 30 times. In Table 2 the best, the worst, median and standard deviation of the results obtained in 30 runs are presented.

As it can be seen, both, PSO and MSA, found the optimal function value for f_1 (sphere). Standard deviation was 0 which means that the optimal solution was found in each run by both algorithms. MSA algorithm found the optimal value for f_5 (different powers function) also with standard deviation 0 while PSO found the optimal value on the presented accuracy but standard deviation was greater then 0, which means that it does not find exactly the optimal solutions. Both algorithms were unable to find even close solutions to the optimal ones for functions f_2, f_3 and f_4. These function are specific and special parameter tuning is necessary and probably more iterations. Even in such conditions, MSA was able to find significantly better solutions compared to the PSO. Similar situation was for the functions f_{14} and f_{15} where even though MSA found rather larger fitness function values compared to the optimal, they were still radically better then the solutions obtained by PSO.

Table 1: 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

MS algorithm obtained better mean and the best fitness function values for f_6 whit lower standard deviation which means that MSA is more stable then the PSO for this function. The worst solution were the same for the both algorithms.

For the functions f_7 to f_{13}, MSA algorithm found better median solutions and in the most cases the best and the worst solutions were better (if not better then the same) compared to the fitness function values obtained by the PSO. Standard deviation was smaller in all cases, except for f_8 and f_{12}. Smaller standard deviation around the worse solutions does not represent an advantage.

The quality of the MSA for functions f_2, f_3,f_4, f_{14} and f_{15} can be probably increased by adjusting the parameters additionally and/or by increasing the maximal iteration numbers.

Based on the result analysis, it can be concluded that MSA has good qualities for solving hard optimization problems and it obtain better results compared to the standard particle swarm optimization.

5 Conclusion

In this paper, recent swarm intelligence algorithm, moth search algorithm was tested on CEC 2013 benchmark functions for unconstrained single objective optimization problems. MSA was compared to the standard particle swarm optimization algorithm and it obtained better results for all test functions. Moreover, MSA found rather good solutions which proves the quality. In further work, MS algorithm

Table 2: Comparison of PSO and MSA

Function		PSO	MSA
f_1	median	-1.400E+03	-1.400E+03
	std	0.000E+00	0.000E+00
	best	-1.40E+03	-1.400E+03
	worst	-1.40E+03	-1.400E+03
f_2	median	3.50E+04	**2.934E+04**
	std	7.36E+04	**8.328E+04**
	best	7.597E+02	**1.853E+02**
	worst	4.755E+05	**4.129E+05**
f_3	median	2.67E+05	**1.284E+05**
	std	1.66E+07	**6.834E+06**
	best	**-1.200E+03**	**-1.158E+03**
	worst	8.251E+07	**1.795E+08**
f_4	median	7.769E+03	**2.359E+03**
	std	4.556E+03	**1.631E+03**
	best	2.454E+02	**1.195E+02**
	worst	1.856E+04	**5.270E+03**
f_5	median	-1.000E+03	-1.000E+03
	std	3.142E-05	**0.000E+00**
	best	-1.000E+03	-1.000E+03
	worst	-1.000E+03	-1.000E+03
f_6	median	-8.902E+02	**-8.256E+02**
	std	4.974E+00	**3.638E+00**
	best	-9.000E+02	**-9.000E+02**
	worst	-8.898E+02	-8.898E+02
f_7	median	-7.789E+02	**-7.582E+02**
	std	1.327E+01	**1.170E+01**
	best	-7.974E+02	**-7.697E+02**
	worst	-7.434E+02	**-7.382E+02**
f_8	median	-6.789E+02	**-6.797E+02**
	std	**6.722E-02**	4.338E-03
	best	-6.789E+02	**-6.797E+02**
	worst	-6.796E+02	**-6.797E+02**
f_9	median	-5.952E+02	**-5.969E+02**
	std	1.499E+00	**1.039E+00**
	best	-5.987E+02	**-5.991E+02**
	worst	-5.929E+02	-5.929E+02
f_{10}	median	-4.999E+02	**-4.999E+02**
	std	2.713E-01	**1.449E-01**
	best	-4.999E+02	**-5.000E+02**
	worst	-4.989E+02	**-4.984E+02**
f_{11}	median	-3.891E+02	**-3.907E+02**
	std	5.658E+00	**4.198E+00**
	best	-3.970E+02	**-3.972E+02**
	worst	-3.731E+02	**-3.781E+02**
f_{12}	median	-2.861E+02	**-2.870E+02**
	std	**6.560E+00**	6.019E+01
	best	-2.970E+02	**-2.971E+02**
	worst	-2.682E+02	-2.623E+02
f_{13}	median	-1.792E+02	**-1.801E+02**
	std	9.822E+00	**8.992E+00**
	best	-1.946E+02	**-1.992E+02**
	worst	-1.523E+022	**-1.617E+02**
f_{14}	median	7.338E+02	**2.914E+02**
	std	1.282E+02	1.282E+02
	best	2.228E+02	**-1.419E+02**
	worst	1.109E+03	**4.990E+02**
f_{15}	median	8.743E+02	**5.695E+02**
	std	2.507E+02	**2.429E+02**
	best	4.372E+02	**4.271E+02**
	worst	1.705E+03	**1.044E+03**

can be applied to some real world optimization problems. Also, it can be adjusted for solving constrained of multi-objective problem or improved by some hybridization, chaotic maps, etc.

References

[1] C., E.R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the 6th international symposium on micromachine and human science, IEEE (1995) 39–43

[2] Marco, D., M., G.L.: Ant colonies for the travelling salesman problem. Biosystems **43** (1997) 73–81

[3] Darvis, Akay, B.: A modified artificial bee colony (ABC) algorithm for constrained optimization problems. Applied Soft Computing **11** (2011) 3021–3031

[4] Brajevic, I., Tuba, M.: An upgraded artificial bee colony algorithm (ABC) for constrained optimization problems. Journal of Intelligent Manufacturing **24** (2013) 729–740

[5] Tuba, M., Bacanin, N., Stanarevic, N.: Adjusted artificial bee colony (ABC) algorithm for engineering problems. WSEAS Transactions on Computers **11** (2012) 111–120

[6] Subotic, M., Tuba, M.: Parallelized multiple swarm artificial bee colony algorithm (MS-ABC) for global optimization. Studies in Informatics and Control **23** (2014) 117–126

[7] Yang, X.S., Deb, S.: Cuckoo search via levy flights. In: Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). (2009) 210–214

[8] Bacanin, N.: Implementation and performance of an object-oriented software system for cuckoo search algorithm. International Journal of Mathematics and Computers in Simulation **6** (2010) 185–193

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

[10] Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. International Journal of Mathematical Modeling and Numerical Optimization **1** (2010) 330–343

[11] Yang, X.S.: A new metaheurisitic bat-inspired algorithm. Studies in Computational Intelligence **284** (2010) 65–74

[12] Tuba, E., Tuba, M., Simian, D.: Adjusted bat algorithm for tuning of support vector machine parameters. In: Congress on Evolutionary Computation (CEC),, IEEE (2016) 2225–2232

[13] Tuba, E., Tuba, M., Simian, D.: Handwritten digit recognition by support vector machine optimized by bat algorithm. In: 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,(WSCG 2016). (2016) 369–376

[14] Tuba, M., Bacanin, N.: Artificial bee colony algorithm hybridized with firefly metaheuristic for cardinality constrained mean-variance portfolio problem. Applied Mathematics & Information Sciences **8** (2014) 2831–2844

[15] Bacanin, N., Tuba, M.: 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, **2014** (2014) 16

[16] Tuba, M.: Asymptotic behavior of the maximum entropy routing in computer networks. Entropy **15** (2013) 361–371

[17] Tuba, M., Brajevic, I., Jovanovic, R.: Hybrid seeker optimization algorithm for global optimization. Applied Mathematics and Information Sciences **7** (2013) 867–875

[18] Tuba, M., Bacanin, N.: Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems. Neurocomputing **143** (2014) 197–207

[19] Wang, G.G.: Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Computing (2016)

[20] Yang, X.S.: Firefly algorithms for multimodal optimization. Stochastic Algorithms: Foundations and Applications, LNCS **5792** (2009) 169–178

[21] Gandomi, A.H., Alavi, A.H.: Krill herd: A new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simulat **17** (2012) 4831–4845

[22] Lin, J.H., Chou, C.W., Yang, C.H., Tsai, H.L., et al.: A chaotic levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. Computer and Information Technology **2** (2012) 56–63

[23] Reynolds, A., Reynolds, D., Smith, A., Svensson, G., Lfstedt, C.: Appetitive flight patterns of male agrotis segetum moths over landscape scales. Journal of Theoretical Biology **245** (2007) 141 – 149

[24] Liang, J., Qu, B., Suganthan, P., Hernández-Díaz, A.G.: Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report **201212** (2013)

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

Eva Tuba
John Naisbitt University
Graduate School of Computer Science
Bulevar umetnosti 29
SERBIA
E-mail: *etuba@ieee.org*

Romana Capor-Hrosik
University of Dubrovnik
Institute for Marine and Coastal Research
Kneza Damjana Jude 12
CROATIA
E-mail: *rcapor@undu.hr*

Adis Alihodzic
University of Sarajevo
Faculty of Mathematics
Zmaj od Bosne 33-35
BOSNIA AND HERZEGOVINA
E-mail: *adis.alihodzic@pmf.unsa.ba*

Marko Beko
Universidade Lusofona de Humanidades e Tecnologias
Campo Grande 376, Lisbon
PORTUGAL
E-mail: *marko@isr.ist.utl.pt*

Raka Jovanovic
Hamad Bin Khalifa University
Doha
QATAR
E-mail: *rjovanovic@qf.org.qa*