Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 1, No. 1 (2024) 12–19.

Paper Type: Original Article

Comparing the Performance of the Wolf Algorithm with Three other Meta-Heuristic Algorithms (Bees, Biogeography-Based, Chicken Swarm)

Hoda Dalili Yazdi[1,*], Reza Tavakkoli Moghaddam[2], Golriz Bolhasani[1]

[1] Department of Industrial Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran; Hdalili59@gmail.com; Tavakolimoghaddam@gmail.com.

[2] Department of Industrial Engineering, University of Tehran, Tehran, Iran; G.bolhasani@yahoo.com.

Citation:

Received: 2 March 2024 *Revised: 19 May 2024* *Accepted: 12 July 2024*	Dalili Yazdi, H., Tavakkoli Moghaddam, R., & Bolhasani, G. (2024). Comparing the performance of the wolf algorithm with three other meta-heuristic algorithms (bees, biogeography-based, chicken swarm). *Metaheuristic Algorithms with Applications, 1 (1), 12-19..*

Abstract

Nowadays, meta-heuristic algorithms have made significant contributions to achieving approximate solutions to optimization problems. It is important to choose a suitable algorithm for each problem, as an algorithm can be appropriate for one type of problem and, at the same time, inappropriate for another one. In this paper, an attempt has been made to compare the Grey Wolf Optimization (GWO) algorithm with 3 modern optimization algorithms (bees algorithm, Biogeography-Based Optimization (BBO) algorithm and Chicken Swarm Optimization (CSO) algorithm). By utilizing 9 criteria functions, the performances of these algorithms in terms of reaching the global optimal point and also the time of reaching have been investigated. In order to make the correct comparison, the selected algorithms are all among the ones which are derived from the foraging behaviors of living organisms.

Keywords: Grey wolf optimization algorithm, Bees algorithm, Biogeography-based optimization algorithm, Chicken swarm optimization algorithm, Criteria functions.

1 | Introduction

In recent years, the use of meta-heuristic algorithms has significantly expanded in solving complex computational problems, and they also have many applications in various branches of science.

Corresponding Author: Hdalili59@gmail.com

Unlike exact optimization methods, these approaches look for points that are as close as possible to the global optimum and are called inexact methods [1].

The Grey Wolf Optimization (GWO) algorithm was introduced by Mirjalili et al. [2], which is based on the hunting behavior of wolves and their hierarchy. In this algorithm, the location of the optimal point is calculated based on the location of the three points (alpha, beta, and delta) (the best points in each iteration).

Simon proposed the Biogeography-Based Optimization (BBO) algorithm, which is based on the mathematical models of biogeography. In the principle of this algorithm, there are two main operators called migration and mutation before optimization. Each individual in the population is evaluated and then goes through migration and mutation steps to reach the global minimum [3].

The algorithm of bees was proposed by Pham et al. [4], which is inspired by the natural search behavior of bees to find the best flower beds (optimal solution).

Leading honey bees communicate the direction of the food source, the distance of the food source, and the richness of the food source to other bees by their body movements. Choosing the best sites and sending more bees to those sites is the method of implementing the bees algorithm (inspired by the behavior of honey bees) [5].

The Chicken Swarm Optimization (CSO) algorithm was presented by Meng et al. [6], which is inspired by the behavior of roosters, chickens and chicks and is proposed to solve practical problems.

2 | Methodology

To compare the performance of the wolf algorithm with the three mentioned algorithms, first, their mathematical models are coded by MATLAB 2014 and then implemented for 9 criteria mathematical functions. In order to familiarize ourselves with these new algorithms, the steps (pseudo-code) of optimization algorithms for wolves and chicken swarms are described in *Figs. 1* and *2*.

```
Xα=the best search agent
Xβ=the second best search agent
Xδ=the third best search agent
while (t < Max number of iterations)
 for each search agent
Update the position of the current search agent by the above equations
end for
Update a, A, and C
Calculate the fitness of all search agents
Update Xα, Xβ, and Xδ
t=t+1
end while
return Xα
```

Fig. 1. Steps (pseudo code) of wolf optimization algorithm [2].

```
Initialize a population of N chickens and define the related parameters;
Evaluate the N chickens' fitness values, t=0;
while (t < Max Generation)
If (t % G == 0)
Rank the chickens' fitness values and establish a hierarchal order in the swarm;
Divide the swarm into different groups and determine the relationship between the chicks and mother
hens in a group; End if
For i = 1 : N
If i == rooster Update its solution/location using equation (1);
 End if
If i == hen Update its solution/location using equation (3);
 End if
If i == chick Update its solution/location using equation (6);
 End if
Evaluate the new solution;
If the new solution is better than its previous one,
update it;
End for
End while
```

Fig. 2. Steps (pseudo code) of CSO algorithm [5].

3 | Results and Discussion

It seems that in order to compare different optimization algorithms, their behaviors should be examined when dealing with solving different mathematical functions.

In this research, 9 criteria functions (Sphere, Schwefel2.22, Schwefel2.21, Rosenbrock, Rastrigin, Ackley, Griewank, Penalty#1 and Penalty#2) were selected [7].

Each of the algorithms has been investigated in different conditions: changing the number of generations (200, 1000, 1500) while keeping the number of the population at 30 and changing the number of the population (20, 30, 50) while keeping the number of generations at 1000.

It should be noted that for each of the specific conditions, 20 executions were performed, and the final answer was obtained from the average of 20 executions in order to reduce the errors.

From the 10 prepared tables, 4 examples of these tables are given below.

I. *Table 1*: fitness functions in repetitions 200 and population number 30.

II. *Table 2*: fitness functions in repetitions 1500 and population number 30.

III. *Table 3*: fitness functions in repetitions 1000 and population number 20.

IV. *Table 4*: fitness functions in repetitions 1000 and population number 50).

Table 1. Fitness functions in repetitions 200 and population number 30.

Functions	GWO	CSO	BBO	Bees
Sphere	1.00E+00	4.11E+04	1.51E+09	8.39E+07
Schwefel2.22	7.50E+00	1.00E+00	1.07E+06	3.14E+05
Schwefel2.21	1.00E+00	1.03E+03	2.85E+02	1.19E+03
Rosenbrock	1.00E+00	2.02E+03	7.52E+00	5.35E+00
Rastrigin	1.04E+04	1.00E+00	5.01E+04	1.18E+05
Ackley	1.00E+00	1.02E+03	8.82E+04	1.70E+05
Griewank	1.00E+00	2.33E+00	1.87E+02	6.88E+01
Penalty#1	1.00E+00	3463034	9.21E+00	3.92E+01
Penalty#2	2.6219	635180.7	1.00E+00	1.79E+00

Table 2. Fitness functions in repetitions 1500 and population number 30.

Functions	GWO	CSO	BBO	Bees
Sphere	1.00E+00	2.44E+27	5.19E+88	1.26E+76
Schwefel2.22	1.77E+03	1.00E+00	8.85E+54	5.76E+48
Schwefel2.21	1.00E+00	1.43E+22	1.61E+21	5.80E+22
Rosenbrock	1.71E+00	2.74E+00	8.82E+00	1.00E+00
Rastrigin	1.00E+00	1.00E+00	#	#
Ackley	1.97E+00	1.00E+00	3.81E+13	2.85E+14
Griewank	1.00E+00	#	#	#
Penalty#1	3.18E+01	5.33E+03	1.00E+00	1.12E+03
Penalty#2	1.19E+14	1.45E+16	4.00E+12	1.00E+00

Table 3. Fitness functions in repetitions 1000 and population number 20.

Functions	GWO	CSO	BBO	Bees
Sphere	1.00E+00	7.41E+11	2.41E+49	1.29E+41
Schwefel2.22	5.79E+06	1.00E+00	1.90E+35	1.71E+31
Schwefel2.21	1.00E+00	3.39E+12	2.70E+11	8.51E+12
Rosenbrock	1.00E+00	3.86E+00	2.25E+01	6.86E+00
Rastrigin	1.00E+00	6.02E+00	1.10E+02	2.10E+02
Ackley	2.90E+00	1.00E+00	6.68E+13	8.69E+14
Griewank	1.32E+00	1.39E+01	3.85E+02	1.00E+00
Penalty#1	1.67E+01	1.26E+07	1.00E+00	1.02E+03
Penalty#2	1.14E+09	2.33E+12	1.11E+08	1.00E+00

Table 4. Fitness functions in repetitions 1000 and population number 50.

Functions	GWO	CSO	BBO	Bees
Sphere	1.00E+00	1.76E+25	5.18E+69	9.58E+60
Schwefel2.22	1.00E+00	1.48E+03	4.24E+39	2.09E+35
Schwefel2.21	1.00E+00	1.35E+18	6.75E+16	1.85E+18
Rosenbrock	2.26E+00	2.41E+00	9.30E+00	1.00E+00
Rastrigin	#	1.00E+00	#	#
Ackley	2.52E+00	1.00E+00	4.01E+13	1.53E+09
Griewank	1.00E+00	2.05E+00	9.18E+01	2.97E+11
Penalty#1	1.56E+01	2.41E+03	1.00E+00	1.11E+02
Penalty#2	7.63E+09	4.75E+10	6.40E+08	1.00E+00

According to the results of the comparison of the time to reach the optimal point in 4 algorithms, the GWO algorithm reaches the solution faster than the other stated algorithms in all conditions. However, regarding the fitness value, the results are different as the number of iterations and population size change.

The calculation results showed in the case of the Sphere function, which is a simple function without local optimal points; if we have many or few repetitions or if we have a large or small population, the Wolf algorithm still gives us a better answer than other algorithms (*Fig. 3*).

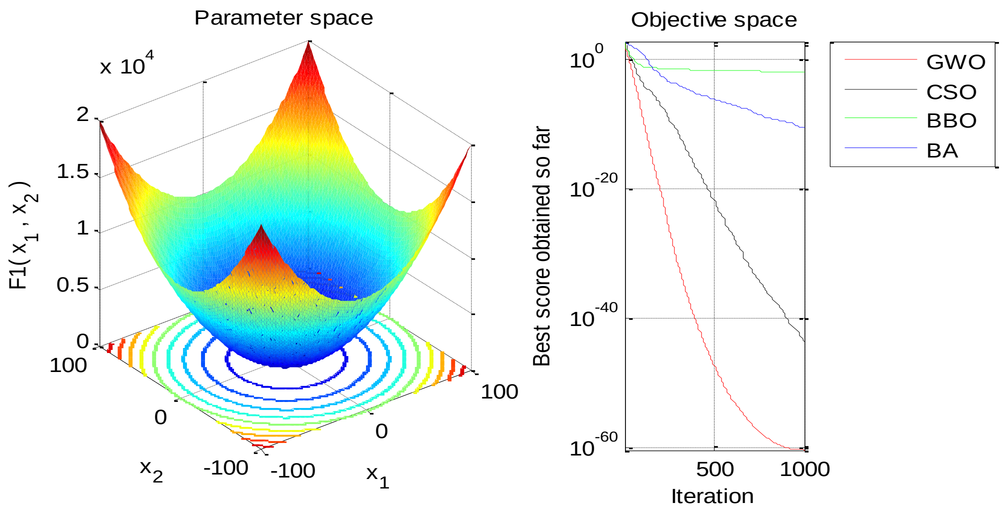

Fig. 3. Comparison of 4 algorithms in Sphere function.

The same rule exists for the Schwefel2.21 function because it is a simple function and does not have local optimal points. As a result, in the case of simple functions that do not have local points, the best algorithm for solving is the wolf algorithm (*Fig. 4*).

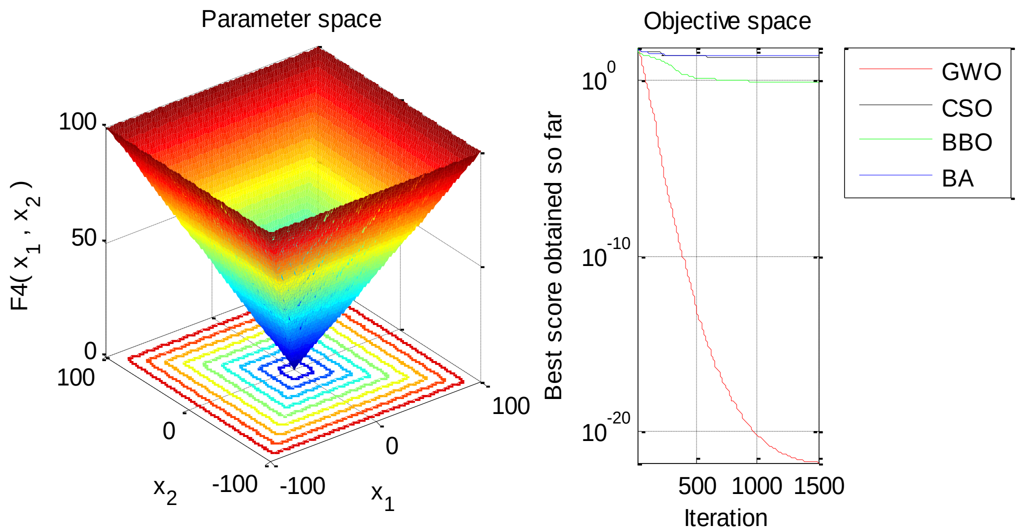

Fig. 4. Comparison of 4 algorithms in Chwefel2.21 function.

In the case of the Schwefel2.22 function, in almost all cases, the chicken swarm algorithm produces a better answer than the rest of the algorithms. This function is a bit more complicated than the Sphere function because it contains both addition and multiplication operations.

On the other hand, this function also has local optimal points. Therefore, the chicken swarm algorithm can be used to solve slightly complex problems with local optimal points (*Fig. 5*).

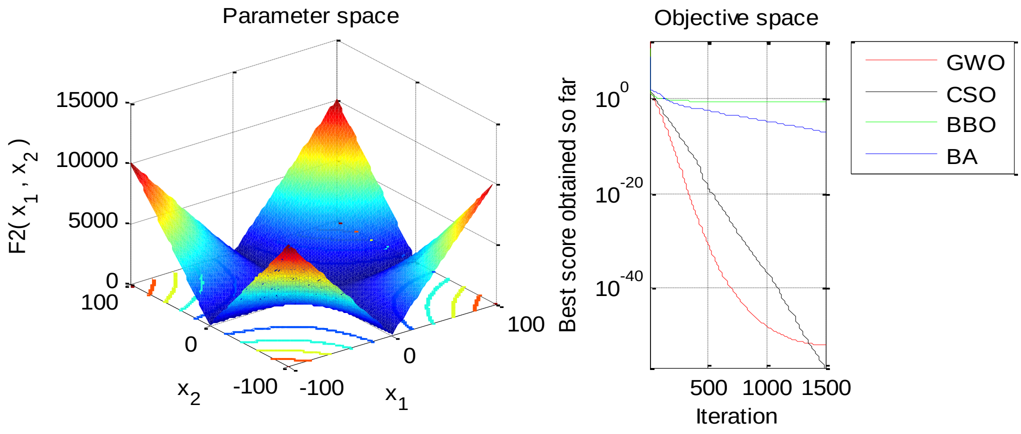

Fig. 5. Comparison of 4 algorithms in Chwefel2.22 function.

The Rosenbrock function is relatively complex, but it does not have local optimal points, and it has no variable multiplication operation. In this function, when we have a population limit or a repetition limit, the gray wolf algorithm still gives us better answers. But when there is no limitation of population and repetitions, the bees algorithm gives us a better answer. The result is that for relatively complex functions where there are no local optimal points, in order to give us a better answer, we need more iterations and more population in the bees algorithm.

In Rastrigin and Ackley functions, in almost all cases, the chicken swarm algorithm gives us a better answer. These functions have many local optimal points, and like the Schwefel2.22 function, which has local optimal points, they get their best solutions from the chicken swarm algorithm (*Fig. 6*).

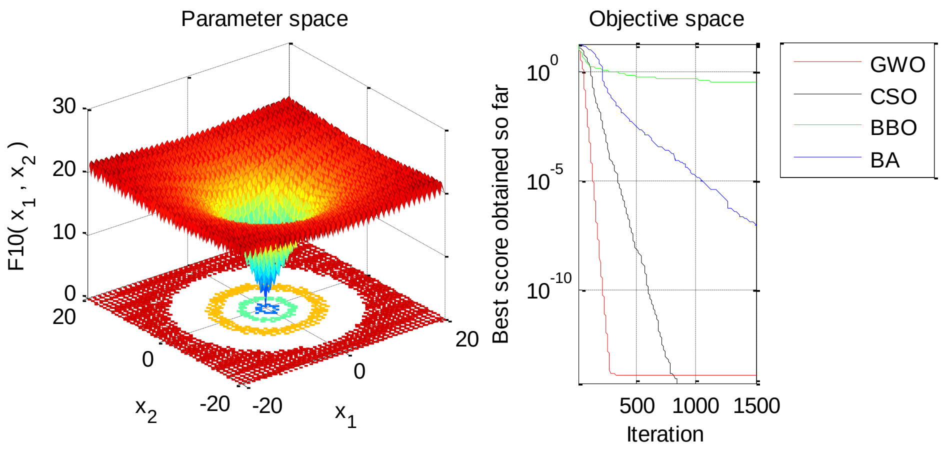

Fig. 6. Comparison of 4 algorithms in Ackley's function.

In the Griewank function, which is a relatively complex function, since it does not have local optimal points, the gray wolf algorithm still gives us the best answers in situations of high or low amounts of population and repetitions.

Penalty#1 and Penalty#2 functions are very complex functions and have local optimal points.

In the Penalty #1 function, the best answers are created by the biogeography algorithm, and in the Penalty #2 function, the best answers are created by the bees algorithm (*Fig. 7*).

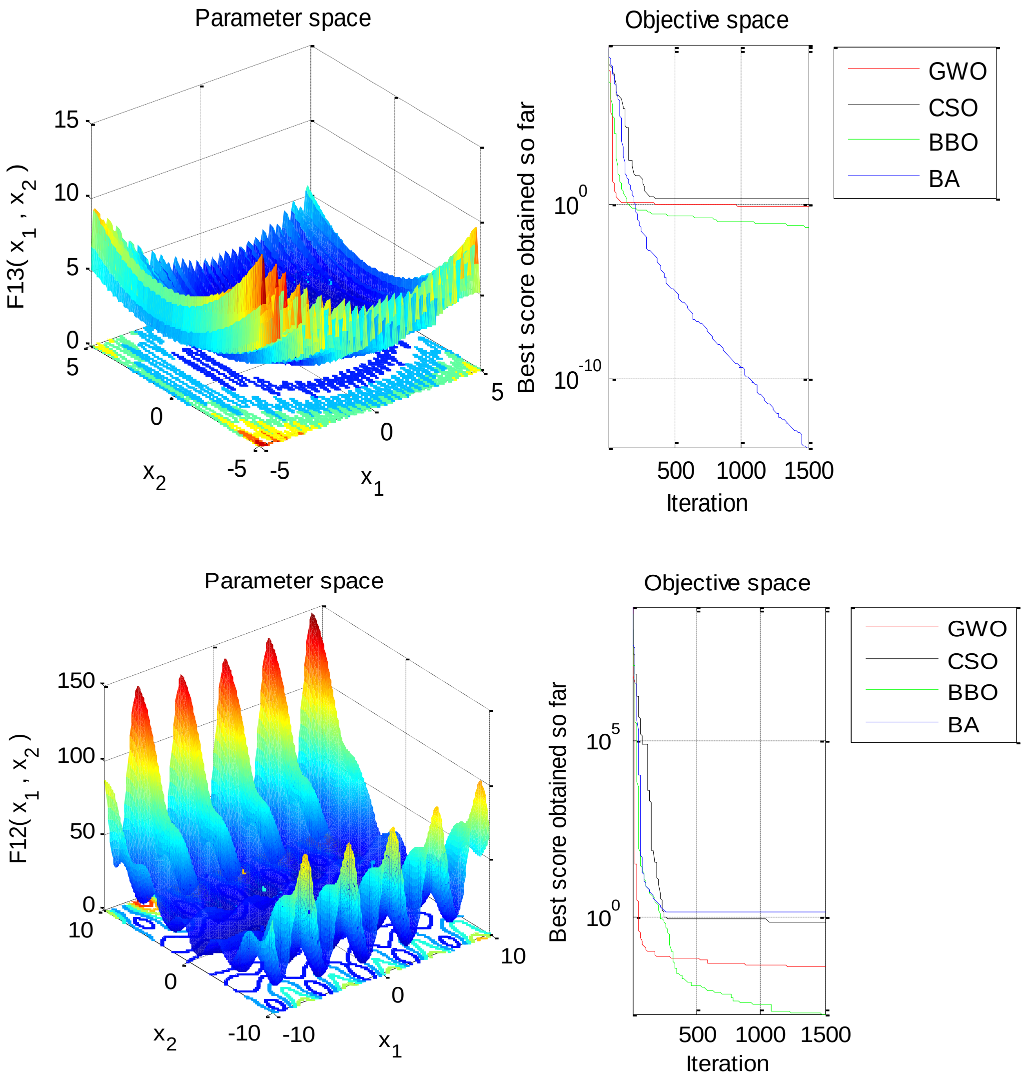

Fig. 7. Comparison of 4 algorithms in Penalty#1 and Penalty#2 functions.

4 | Conclusion

The result is that the gray wolf algorithm is usually caught in the trap of local optimal points, so it is not suitable for problems that have local optimal points. However, in other problems, due to the high speed of convergence of this algorithm, it is a suitable option for solving.

Unlike the wolf algorithm, the chicken swarm algorithm does not fall into the trap of local optima, and it seems that using the CSA advantages is useful in modifying the wolf algorithm. For complex problems, bees' algorithms and biogeography have the ability to reach optimal points, and they are suitable for such issues.

References

[1] Tavakoli Moghadam, R., Nourozi, N., Kalami, S. M., & Salamat Bakhsh, A. (2013). *Meta heuristic Algorithms*. Islamic Azad University. (**In Persian**). https://www.fadakbook.ir/product/19594/

[2] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[3] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, *12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004

[4] Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The bees algorithm—a novel tool for complex optimisation problems. In *Intelligent production machines and systems* (pp. 454–459). Elsevier.

[5] Xing, B., & Gao, W.-J. (2014). *Innovative computational intelligence: a rough guide to 134 clever algorithms* (Vol. 62). Springer.

[6] Meng, X., Liu, Y., Gao, X., & Zhang, H. (2014). A new bio-inspired algorithm: chicken swarm optimization. *Advances in swarm intelligence: 5th international conference, ICSI 2014, Hefei, China, proceedings, part I 5* (pp. 86-94). Springer International Publishing. https://doi.org/10.1007/978-3-319-11857-4_10%0A%0A

[7] Dixon, L. C. W. (1978). The global optimization problem: an introduction. *Towards global optimiation 2*, 1–15. https://cir.nii.ac.jp/crid/1573105974275467776