



Solving optimization problems using black hole algorithm

Masoum Farahmandian ^{1*}, Abdolreza Hatamlou ²

¹ Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

² Department of Computer Engineering, Khoy Branch, Islamic Azad University, Khoy, Iran

*Corresponding author E-mail: gmasafat@gmail.com

Copyright © 2015 Masoum Farahmandian, Abdolreza Hatamlou. This is an open access article distributed under the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Various meta-heuristic optimization approaches have been recently created and applied in different areas. Many of these approaches are inspired by swarm behaviors in the nature. This paper studies the solving optimization problems using Black Hole Algorithm (BHA) which is a population-based algorithm. Since the performance of this algorithm was not tested in mathematical functions, we have studied this issue using some standard functions. The results of the BHA are compared with the results of GA and PSO algorithms which indicate that the performance of BHA is better than the other two mentioned algorithms.

Keywords: Optimization, Meta-Heuristic Optimization, Black Hole Algorithm, Particle Swarm Optimization, Genetic Algorithm.

1. Introduction

Optimization algorithms are powerful methods for solving complex problems [1]. Optimization approaches are divided into exact algorithms [2] and approximate algorithms [3-5]. Exact approaches are able to find the exact optimal answer; however, these approaches are not useful in NP-hard [2], [6-7] problems since their solving time for these NP-hard problems increases exponentially. Approximate algorithms are able to find good answers (near to optimal solutions) for NP-hard in a short time. They are divided into two groups of heuristic algorithms [8-10] and meta-heuristic algorithms[11]. Heuristic approaches create suitable and good solutions which normally are not the best solution. Meta-heuristic approaches have been formed according to inspiration by nature, physics and human being. In recent years, many of these algorithms and their improved algorithms have been successfully applied to various problems of engineering optimization [12-16]. A common feature in meta-heuristic approaches is that they combine rules and randomness to imitate natural phenomena. These phenomena include the biological evolutionary process (e.g., the Genetic Algorithm (GA)[17-18] and the Differential Evolution (DE) [12-13]), animal behavior (e.g., Particle Swarm Optimization (PSO) [14] and Ant Colony Algorithm (ACA) [15-16]), and the physical annealing process (e.g., Simulated Annealing (SA) [1, 19]). These algorithms are one of the approximate optimization approaches that have mechanism of departing from local optimum.

Many of the meta-heuristic algorithms have been successfully applied to various engineering optimization problems over the recent years [12-16]. In order to get suitable solutions, they have better performance than conventional calculations methods for some complicated and difficult real-world optimization problems. Two main problems of heuristic approaches are their existence in local optimum and their incapability to be used in various problems. While, Meta-heuristic approaches are presented to solve these problems[20]. Some of the algorithms which are used to categorize the meta-heuristic approaches are as follows [21-22]:

- Population-based algorithms [23].
- nature-inspired and non-nature-inspired algorithms [24-25] (most of the meta-heuristic approaches are inspired by nature.).
- with-memory and without-memory algorithms [26].
- exact and approximate algorithms [22], [27].

Solving optimization problems based on the BHA is proposed here. Then, the obtained results are compared with the two optimization GA and PSO algorithm. This paper is organized as follows: in section 2, the BHA and its characteristics are reviewed. In section 3, experimented and comparative studies are presented and experimental results are shown. Finally the section 4 is the conclusion for the performed experiments.

2. Black hole algorithm (BHA)

BHA [28] is a population-based and one of the newest approaches that has been recently created and used successfully for solving optimization problems. However, its performance and operation has not yet been tested on standard functions. This research studies the performance of BHA over the standard functions and then compares it with other optimization algorithms.

The population of candidate solutions (stars) in this algorithm is generated randomly from the existed points in research space. After initialization, the fitness values of the population are evaluated and the best candidate –which has the best fitness values –is introduced as black hole and the other stars are selected as normal stars. Then, all the stars commence moving towards the black hole and the black hole absorbs the stars around it. The movement formal of stars towards black hole is as follows [28]:

$$x_i(t+1) = x_i(t) + \text{rand} * (x_{\text{BH}} - x_i(t)) \quad i = 1, 2, 3, \dots, N \quad (1)$$

Where $x_i(t+1)$ and $x_i(t)$ are the locations of the i_{th} star at iterations $t+1$ and t . Rand is a random number between zero and one. x_{BH} is the location of the BH in our search space. N is the number of candidate solutions (stars) [28].

After the movement of the star to the new position (towards the black hole), if its fitness value is better than the value of black hole, the star is selected as the black hole. Then this algorithm continue with the black hole in the new location and stars start moving towards this new black hole.

Moreover, there is the probability of crossing the event horizon (star's distance with black hole) during stars movement towards the black hole. Candidate solution (star) that crosses the event horizon of the black hole will be swallowed by the black hole. Then a new star following the swallowed one is generated and distributed randomly in the search space [28-29]. This generation is to keep the number of stars (candidate solutions) constant. When all the stars moved once, next iteration takes place.

The radius of the event horizon (R) is formulated as follows[28]:

$$R = \frac{f_{\text{BH}}}{\sum_{i=1}^N f_i} \quad (2)$$

Where f_{BH} is the fitness value of the black hole and N is the number of candidate solutions (stars) and f_i is the fitness value of the i_{th} star. when the star's distance with black hole is less than a defined radius (R), this star is swallowed by the black hole [28-29].

According to the above explanation the main step in BHA is summarized as follows[28].

3. Experiments

In order to study the performance of BHA on unconstrained optimization problems, we use nineteen standard test functions, which are benchmarks from[30]. The evaluated results are compared with the prominent algorithms (GA and PSO) reported in[30]. Related test functions are shown in tables 1, 2, 3. Tables 1, 2, 3 give the details of the test functions, including dimensions, their equations, optimal values and the domains. These functions encoded by MATLAB R7 with PC system, Pentium IV, CPU 3 GHZ and 1GB RAM is performed and tested.

Each of these test functions have been solved by black hole for 30 times. The starting values of the variables for each problem are selected randomly for all runs from the solution space. The results found by black hole such as the best average function value, numbers of functions evaluation and solution time in seconds are recorded in table 4, 5, 6, whereas for the other algorithms only the function value and numbers of function evaluation are given, because the solution times for some algorithms are not given. As it is deduced from table 4, 5, 6, the black hole is successful while finding the optimum solution of the given functions. These results suggest that BHA is better than two other algorithms i.e. GA and PSO.

In this section, a comparative study has been carried out to evaluate the proposed approach concerning quality and performance of the solution. On one hand, evolutionary techniques suffer from the quality of solution; therefore, BH approach is used to increase the solution quality by combining the features of two heuristic algorithms. On the other hand, unlike classical techniques it doesn't search from a single point but search from a population of points. Therefore, it provides a globally optimal solution. In addition, this approach uses only the objective function information, not derivatives or other auxiliary knowledge. Therefore, it can deal with the non-continuous, non-smooth and non-differentiable functions which are actually existed in practical optimization problems. Another peculiarity is that the

simulation results prove superiority of the BHA to those reported in the literature, where our proposed approach is completely better than the other approaches. So, the BHA is quite competitive when it is compared with the other existing approaches. Finally, the fact of using the BH approach to handle complex problems of realistic dimensions is approved due to procedure simplicity.

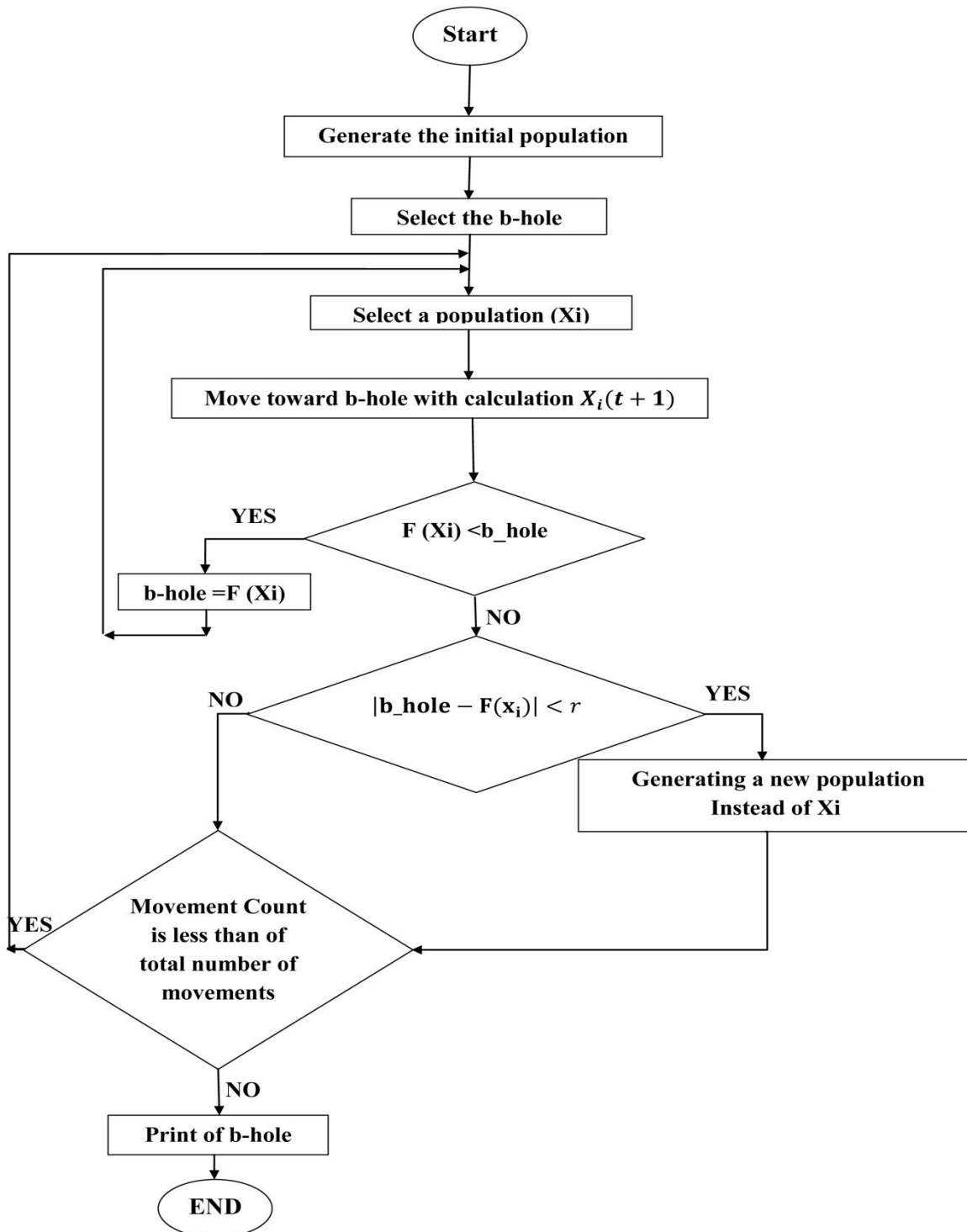


Fig. 1: The Flowchart of BHA

2.1. Benchmark functions

The used functions in this article are shown in tables 1, 2, 3. In these tables S-D is the used limit for function and n is the number of that function. The minimum value (f_{opt}) for all functions in table 1 and 2 is zero except F_8 which is a minimum value of $-418.9829 \times n$. The Optimum location (X_{opt}) also for all functions in table 1 and 2, are in $[0]^n$, except for F_5 and F_{12} with X_{out} in $[1]^n$ and F_8 in $[420.96]^n$. A detailed description of the functions of Table 3 is given in Appendix A.

Table 1: Unimodal Test Functions.

Test functions	S-D
$F_1(x) = \sum_{i=1}^n x_i^2$	$[-100,100]^n$
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]^n$
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100,100]^n$
$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100,100]^n$
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$
$F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100,100]^n$
$F_7(x) = \sum_{i=1}^n ix_i^4 + random[0,1]$	$[-1.28,1.28]^n$

Table 2: Multimodal Test Function.

Test functions	S-D
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500,500]^n$
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$
$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^n$
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$
$F_{12}(x) = -0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + i=1 nu(x_i, 5, 100, 4)$	$[-50,50]^n$

Table 3: Multimodal Test Functions with Fix Dimension.

Test functions	S-D
$F_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^25 (x_i - a_{ij})^6} \right)^{-1}$	$[-65.536,65.536]^2$
$F_{14}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5,5]^4$
$F_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^2$
$F_{16}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	$[-5,10] \times [0,15]$
$F_{17}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 * 18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	$[-5,5]^2$
$F_{18}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	$[0,1]^3$
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	$[0,1]^6$

After the evaluation of algorithms by these functions the following results are achieved:

2.1.1. Unimodal high-dimensional functions

The current functions in table 1 (F1 to F7) are in the category of unimodal high-dimensional Function. Each of these functions was performed for 30 times and their Average best so far and Median best so far and Average mean fitness are registered in table 4. The values of this tables shows that the performance of BH approach in these functions is better than GA and PSO.

Table 4: Minimization Result of Functions in Table 1 with N=30.

		GA[1]	PSO[1]	BHA
F1	Average best so far	23.13	1.8×10^{-3}	9.57×10^{-9}
	Median best so far	21.87	1.8×10^{-3}	2.94×10^{-4}
	Average mean fitness	23.45	1.8×10^{-2}	1.34×10^{-2}
F2	Average best so far	1.07	2.0	5.49×10^{-3}
	Median best so far	1.13	1.9×10^{-3}	8.81×10^{-3}
	Average mean fitness	1.07	2.0	9.17×10^{-3}
F3	Average best so far	$5.6 \times 10^{+3}$	$4.1 \times 10^{+3}$	$4.7 \times 10^{+1}$
	Median best so far	$5.6 \times 10^{+3}$	$2.2 \times 10^{+3}$	$1.1 \times 10^{+2}$
	Average mean fitness	$5.6 \times 10^{+3}$	$2.9 \times 10^{+3}$	$1.1 \times 10^{+2}$
F4	Average best so far	11.78	8.1	4.2×10^{-2}
	Median best so far	11.94	7.4	9.42×10^{-2}
	Average mean fitness	11.78	23.6	1.1×10^{-1}
F5	Average best so far	$1.1 \times 10^{+3}$	$3.6 \times 10^{+4}$	17.4

	Median best so far	$1.1 \times 10^{+3}$	$1.7 \times 10^{+3}$	57.16
	Average mean fitness	$1.1 \times 10^{+3}$	$3.7 \times 10^{+4}$	61.19
	Average best so far	24.01	1.0×10^{-3}	3.4×10^{-4}
F6	Median best so far	24.55	6.6×10^{-3}	6.3×10^{-4}
	Average mean fitness	24.52	0.02	7.1×10^{-4}
	Average best so far	0.06	0.04	0.02
F7	Median best so far	0.06	0.04	0.07
	Average mean fitness	0.56	1.04	0.07

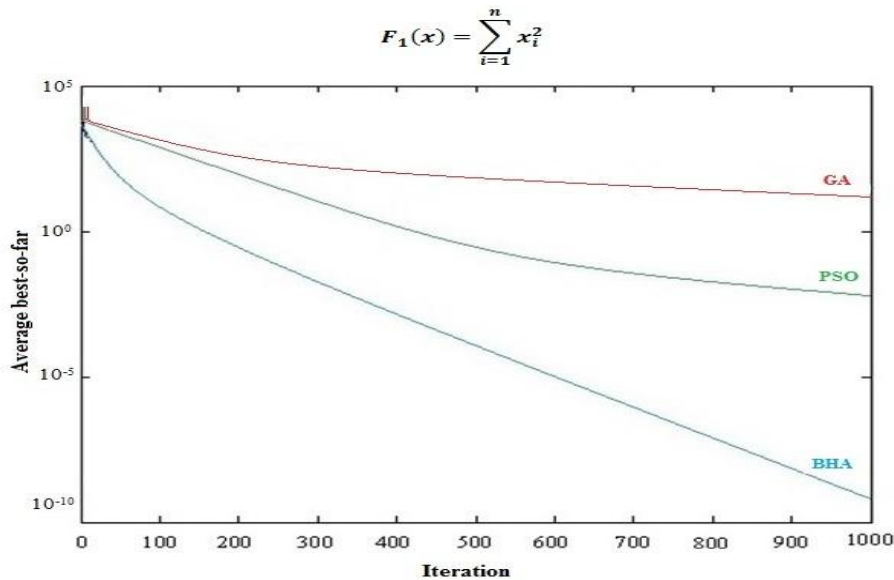


Fig. 2: Comparison of Performance of GA, PSO and BHA for Minimization of $F_1(X)_n$ with $N=30$ and Iteration=1000.

2.1.2. Multimodal high-dimensional functions

These functions (F8 to F12) are in the category of Multimodal high-dimensional Functions. These functions are one of those difficult functions in the field of optimization. Each of these functions was performed for 30 times and their Average best so far and Median best so far and Average mean fitness were registered in table 5. The achieved values in table 5 show that the performance of BH approach in these functions especially in functions F10 and F11 is better than GA and PSO.

Table 5: Minimization Result of Functions in Table 2 with $N=30$.

		GA[1]	PSO[1]	BHA
F8	Average best so far	$-1.2 \times 10^{+4}$	$-9.8 \times 10^{+3}$	$-6.8 \times 10^{+3}$
	Median best so far	$-1.2 \times 10^{+4}$	$-9.8 \times 10^{+3}$	$-5.4 \times 10^{+3}$
	Average mean fitness	$-1.2 \times 10^{+4}$	$-9.8 \times 10^{+3}$	$-5.5 \times 10^{+3}$
F9	Average best so far	5.90	55.1	14.9
	Median best so far	5.71	56.6	28.3
	Average mean fitness	5.92	72.8	28.9
F10	Average best so far	2.13	9.0×10^{-3}	3.8×10^{-3}
	Median best so far	2.16	6.0×10^{-3}	6.6×10^{-3}
	Average mean fitness	2.15	0.02	7.4×10^{-3}
F11	Average best so far	1.16	0.01	3.8×10^{-4}
	Median best so far	1.14	0.0081	9.3×10^{-3}
	Average mean fitness	1.16	0.055	1.4×10^{-2}
F12	Average best so far	0.081	3.1×10^{-18}	4.7×10^{-4}
	Median best so far	0.032	2.2×10^{-23}	9.3×10^{-4}
	Average mean fitness	0.081	$4.8 \times 10^{+5}$	1.0×10^{-3}

2.1.3. Multimodal low dimensional functions

The functions F13 to F19 are in the category of Multimodal low - dimensional Functions. Each of these functions was performed for 30 times and their Average best so far and Median best so far and Average mean fitness were registered in table 6. The achieved values in these functions suggest that BH approach in these functions are very near and better than the values of GA and PSO.

Table 6: Minimization Result of Functions in Table 2 with N=30.Maximum Number of Iterations=500.

		GA[1]	PSO[1]	BHA
F13 n=2	Average best so far	0.998	0.998	0.998
	Median best so far	0.998	0.998	0.998
	Average mean fitness	0.998	0.998	0.998
F14 n=4	Average best so far	4.0×10^{-3}	2.8×10^{-3}	3.2×10^{-4}
	Median best so far	1.7×10^{-3}	7.1×10^{-4}	4.7×10^{-4}
	Average mean fitness	4.0×10^{-3}	215.60	6.8×10^{-4}
F15 n=2	Average best so far	-1.0313	-1.0316	-1.0316
	Median best so far	-1.0315	-1.0316	-1.0316
	Average mean fitness	-1.0313	-1.0316	-1.0316
F16 n=2	Average best so far	0.3996	0.3979	0.3978
	Median best so far	0.3980	0.3979	0.3978
	Average mean fitness	0.3996	2.4112	0.3978
F17 n=2	Average best so far	5.70	3.0	1.0
	Median best so far	3.0	3.0	1.0
	Average mean fitness	5.70	3.0	1.0
F18 n=3	Average best so far	-3.8627	-3.8628	-3.8794
	Median best so far	-3.8628	-3.8628	-3.8794
	Average mean fitness	-3.8627	-3.8628	-3.8794
F19 n=6	Average best so far	-3.3099	-3.2369	-3.3225
	Median best so far	-3.3217	-3.2031	-3.3224
	Average mean fitness	-3.3098	-3.2369	-3.2817

4. Conclusion

Since even a minor shift in the answer of a problem can change the cost and the duration of the project to a large extent, the solving optimization problems is one of the most significant parts of implementation of research projects. In order to get this goal, using meta-heuristic algorithms will be suitable because they are suitable both in time and cost. In this regard, BHA is introduced as a new and successful algorithm since it's based on population and has less complexity in comparison with other algorithms. Because the performance of BHA in the field standard functions had not been studied and tested before, we studied the effect of this algorithm on the mathematic issues and compared the results with GA and PSO algorithms. The results suggested that BHA has better performance than other algorithms. Therefore, this algorithm can be used for optimization of different and complex projects. Due to this reason, this algorithm will be one of the known and widely used algorithms in near future.

Appendix A

Table A.1: a_{ij} in F_{13}

	J=1	J =2	J =3	J =4	J =5	J =6	J =7	J =8	J =9	J =10	J =21	J =22	J =23	J =24	J =25
I=1	-32	-16	0	16	32	-32	-16	0	16	32	-32	-16	0	16	32
I=2	-32	-32	-32	-32	-32	-16	-16	-16	-16	-16	32	32	32	32	32

Table A.2: a_i and b_j in F_{14} .

I	1	2	3	4	5	6	7	8	9	10	11
a_i	0.1957	0.1957	0.1735	0.1600	0.0844	0.0627	0.0456	0.0342	0.0342	0.0235	0.0246
b_j	0.25	0.5	1	2	4	6	8	10	12	14	16

Table A.3: a_j and c_i in F_{18} .

12	a_{i1}	a_{i2}	a_{i3}	c_i
1	3	10	30	1
2	0.1	10	35	1.2
3	3	10	30	3
4	0.1	10	30	3.2

Table A.4: P_{ij} in F_{18} .

i	P_{i1}	P_{i2}	P_{i3}
1	0.3689	0.1170	0.2673
2	0.4699	0.4387	0.7470
3	0.1091	0.8732	0.5547
4	0.03815	0.5743	0.8828

Table A.5: a_{ij} and c_i in F_{19} .

I	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	c_i
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	17	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2

Table A.6: P_{ij} in F_{19} .

I	P_{i1}	P_{i2}	P_{i3}	P_{i4}	P_{i5}	P_{i6}
1	0.131	0.169	0.556	0.012	0.828	0.588
2	0.232	0.413	0.830	0.373	0.100	0.999
3	0.234	0.141	0.352	0.288	0.304	0.665
4	0.404	0.882	0.873	0.574	0.109	0.038

References

- [1] Rashedi, E., H. Nezamabadi-Pour, and S. Saryazdi, *GSA: a gravitational search algorithm*. Information sciences, 2009. 179(13): p. 2232-2248. <http://dx.doi.org/10.1016/j.ins.2009.03.004>.
- [2] Woeginger, G.J., *Exact algorithms for NP-hard problems: A survey*, in Combinatorial Optimization—Eureka, You Shrink! 2003, Springer. p. 185-207. http://dx.doi.org/10.1007/3-540-36478-1_17.
- [3] Pozzi, L., K. Atasu, and P. Jenne, *Exact and approximate algorithms for the extension of embedded processor instruction sets*. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2006. 25(7): p. 1209-1229. <http://dx.doi.org/10.1109/TCAD.2005.855950>.
- [4] Park, H. and K. Shim, *Approximate algorithms for k-anonymity*. in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 2007: ACM. <http://dx.doi.org/10.1145/1247480.1247490>.
- [5] Aksoy, L., et al., *Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications*. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2008. 27(6): p. 1013-1026. <http://dx.doi.org/10.1109/TCAD.2008.923242>.
- [6] Klein, P.N. and N.E. Young, *Approximation algorithms for NP-hard optimization problems*. in *Algorithms and theory of computation handbook*. 2010: Chapman & Hall/CRC.
- [7] Guo, H. and W.H. Hsu, *machine learning approach to algorithm selection for \mathcal{NP}-hard optimization problems: a case study on the MPE problem*. Annals of Operations Research, 2007. 156(1): p. 61-82. <http://dx.doi.org/10.1007/s10479-007-0229-6>.
- [8] Quiroz, M., et al., *Improving the Performance of Heuristic Algorithms Based on Exploratory Data Analysis*, in *Recent Advances on Hybrid Intelligent Systems*. 2013, Springer. p. 361-375. http://dx.doi.org/10.1007/978-3-642-33021-6_29.
- [9] Do Prado, P.F., et al. *A Performance Evaluation Study for QoS-aware Web Services Composition Using Heuristic Algorithms*. in *ICDS 2013, The Seventh International Conference on Digital Society*. 2013.
- [10] Hirsch, E., et al., *Optimal heuristic algorithms for the image of an injective function*. Journal of Mathematical Sciences, 2013. 188(1): p. 7-16. <http://dx.doi.org/10.1007/s10958-012-1102-y>.
- [11] Lukasiewicz, M., et al. *Opt4J: a modular framework for meta-heuristic optimization*. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. 2011: ACM.
- [12] Mashinchi, M.H., M.A. Orgun, and W. Pedrycz, *Hybrid optimization with improved tabu search*. Applied Soft Computing, 2011. 11(2): p. 1993-2006. <http://dx.doi.org/10.1016/j.asoc.2010.06.015>.
- [13] Serrurier, M. and H. Prade, *Improving inductive logic programming by using simulated annealing*. Information sciences, 2008. 178(6): p. 1423-1441. <http://dx.doi.org/10.1016/j.ins.2007.10.015>.
- [14] Marinakis, Y., M. Marinaki, and G. Dounias, *Particle swarm optimization for pap-smear diagnosis*. Expert Systems with Applications, 2008. 35(4): p. 1645-1656. <http://dx.doi.org/10.1016/j.eswa.2007.08.089>.
- [15] Van Sickle, J.H., K.Y. Lee, and J.S. Heo, *Differential evolution and its applications to power plant control*. In *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*. 2007: IEEE.
- [16] Qing, A., *Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems*. Geoscience and Remote Sensing, IEEE Transactions on, 2006. 44(1): p. 116-125. <http://dx.doi.org/10.1109/TGRS.2005.859347>.
- [17] Soleimanian, F., I. Maleki, and M. Farahmandian, *New Approach for Solving Dynamic Travelling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization*. Methods, 2012. 53(1).
- [18] Castillo, O., et al., *Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot*. Information sciences, 2012. 192: p. 19-38. <http://dx.doi.org/10.1016/j.ins.2010.02.022>.
- [19] Wolpert, D.H. and W.G. Macready, *No free lunch theorems for optimization*. Evolutionary Computation, IEEE Transactions on, 1997. 1(1): p. 67-82. <http://dx.doi.org/10.1109/4235.585893>.
- [20] Yang, X.-S., *Nature-inspired metaheuristic algorithms*. 2010: Luniver Press.
- [21] Pooranian, Z., et al., *An efficient meta-heuristic algorithm for grid computing*. Journal of Combinatorial Optimization, 2013: p. 1-22.
- [22] Talbi, E.-G., *Metaheuristics: from design to implementation*. Vol. 74. 2009: John Wiley & Sons. <http://dx.doi.org/10.1002/9780470496916>.
- [23] Lin, Y.-C. and M. Middendorf, *Simple probabilistic population based optimization for combinatorial optimization*. In *Swarm Intelligence (SIS), 2013 IEEE Symposium on*. 2013: IEEE.
- [24] Yang, X.-S., *Metaheuristic Optimization: Nature-Inspired Algorithms and Applications*, in *Artificial Intelligence, Evolutionary Computing and Metaheuristics*. 2013, Springer. p. 405-420. http://dx.doi.org/10.1007/978-3-642-29694-9_16.
- [25] Fister Jr, I., et al., *A Brief Review of Nature-Inspired Algorithms for Optimization*. arXiv preprint arXiv:1307.4186, 2013.
- [26] Ninin, J. and F. Messine, *A metaheuristic methodology based on the limitation of the memory of interval branch and bound algorithms*. Journal of Global Optimization, 2011. 50(4): p. 629-644. <http://dx.doi.org/10.1007/s10898-010-9531-y>.
- [27] Brusco, M.J. and D. Steinley, *Exact and approximate algorithms for variable selection in linear discriminant analysis*. Computational Statistics & Data Analysis, 2011. 55(1): p. 123-131. <http://dx.doi.org/10.1016/j.csda.2010.05.027>.
- [28] Hatamlou, A., *Black hole: A new heuristic optimization approach for data clustering*. Information sciences, 2012: p. 175-184.
- [29] Chen, S.-M. and Y.-C. Chang, *Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques*. Information sciences, 2010. 180(24): p. 4772-4783. <http://dx.doi.org/10.1016/j.ins.2010.08.026>.
- [30] Chen, S.-M. and C.-Y. Chien, *Parallelized genetic ant colony systems for solving the traveling salesman problem*. Expert Systems with Applications, 2011. 38(4): p. 3873-3883. <http://dx.doi.org/10.1016/j.eswa.2010.09.048>.