

Modified Black Hole Algorithm with Genetic Operators

Saber Yaghoobi, Hamed Mojallali

*Electrical Engineering Department, Faculty of Engineering, University of Guilan,
Rasht, PO Box 3756, Guilan Province, Iran
Emails: saber.yaghoobi@gmail.com, mojallali@guilan.ac.ir*

Received 3 November 2015

Accepted 12 March 2016

Abstract

In this paper, a modified version of nature-inspired optimization algorithm called Black Hole has been proposed. The proposed algorithm is population based and consists of genetic algorithm operators in order to improve optimization results. The proposed method enhances Black Hole algorithm performance by searching space with more diversity. The modified Black Hole algorithm has been applied to a well-known benchmark. The experimental results show that the modified Black Hole algorithm outperforms compared to some prominent optimization algorithms.

Keywords: Black Hole, Nature-inspired optimization, Metaheuristic algorithm, Benchmarking.

1. Introduction

Optimization problems have become a major field of interest for researchers in different propensities, and created an immense bunch of movement widely used in engineering applications. Due to complexity and nonlinearity of many engineering optimization problems, classic optimization approaches may tend to fail. Nature and bio inspired algorithms seem superior to their classic counterparts and have found specific places between the newfangled algorithms [1]. The effectiveness of these algorithms had been proven in many fields such as industry, hardware design, urban design, routing, image processing, project scheduling, Controller design, robots path planning, data clustering and etc. [2]

The Black Hole (BH) Algorithm is one of the optimization techniques inspired by the behavior of the real black holes. The black hole terminology is used for the first time in [3] to improve the convergence rate and efficiency of the PSO algorithm. Then, a new simple version of the BH algorithm was proposed in [4] for data clustering. In recent years, the BH algorithm and its modified versions have been used to solve optimization and engineering problems [5-18].

The aim of this paper is to cope drawbacks of the BH algorithm proposed in [4] such as getting trapped in local minima, and to provide the ability to solve both high and low dimensional problems. For this purpose, a modified Black Hole (MBH) algorithm is proposed in which the genetic programming and swarm intelligence has been employed concurrently to solve optimization problems. At each iteration, a particle with the best cost named as the Black Hole, and other particles called stars start moving towards the black hole with a random generated angle to cover the search space and find the best local minima. Similar to real black hole phenomena, a star is swallowed by the black hole if its distance to the black hole reaches the minimum. Also, the genetic operators called crossover and mutation are applied to the obtained space in order to improve the particles cost. The main modifications are as follows:

In standard Black Hole Algorithm, stars move directly toward to the black hole. But, in the proposed algorithm, stars move toward to the black hole with a restricted angle between $-\pi/4$ and $+\pi/4$, so the search space has been enlarged. The swallowing formula is changed to avoid the loss of stars close to global optimum when there are a lot of local optimums. The

random numbers are selected between 0 and 2. Therefore, the proposed algorithm is able to scour the area around the black hole on both sides to find the best probable optimum. Some genetic algorithm operators (mutation and crossover) have been used in the proposed algorithm. The genetic operators are applied to improve the particles cost.

The rest of this paper is organized as follows: the black hole phenomena and standard optimization algorithm are presented in section 2. Section 3 describes the proposed MBH algorithm. A detailed comparison of the MBH algorithm and some prominent optimization algorithms are given in section 4. The paper is concluded in section 5.

2. Standard Black Hole Optimization Algorithm

A black hole is a geometrically defined region of space-time exhibiting such strong gravitational effects that nothing, including particles and electromagnetic radiation such as light, can escape from inside it [19]. The theory of a star becoming invisible was formulated by John Michell and Pierre Laplace in the eighteenth-century. Then, John Wheeler named the mass collapsing phenomenon as a black hole in 1967 [4].

The size of a black hole, determined by the radius of event horizon or Schwarzschild radius, is roughly proportional to the mass M through

$$R = \frac{2GM}{c^2} \quad (1)$$

where G is the gravitational constant and c is the speed of light.

The BH algorithm is inspired by the behavior of real black holes. Like other population-based optimization algorithms, the stars as a randomly generated population of candidate solutions are placed in the search space. The movement of stars towards the black hole is formulated as follows [4]:

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

where $x_i(t)$ and $x_i(t+1)$ are the locations of the i th star at iterations t and $t+1$, respectively. x_{BH} is the location of the black hole in the search space. $rand$ is a random number in the interval $[0, 1]$. In order to avoid trapping in local minima, if any star reaches a certain distance to the black hole it will be eliminated as in real black hole

phenomena. Therefore, a new particle is created in the space randomly. This distance is calculated by the following formula [4]:

$$R = \frac{f_{BH}}{\sum_{i=1}^N f_i} \quad (3)$$

where f_{BH} is the fitness value of the black hole, f_i is the fitness value of the i th star and N is the number of stars.

3. Proposed Modified Black Hole Algorithm (MBH)

Flowchart of the proposed modified black hole algorithm (MBH) is shown in Fig. 1. As noted, the modified black hole is a population-based optimization algorithm. A random population is generated and values of the cost function are calculated for all candidate solutions. Afterwards, exploration and exploitation sections are employed to analyze the search space, yet with the minimum computational efforts to reach the best point, independent of either being the minimum or the maximum. First, the random population is distributed in the search space. Then, the point with the lowest cost among others is picked to be the black hole, while the others are known as the stars. As in real black hole, due to its particular characteristics, the air converges at the ground level and all the stars try to reach the black hole. Here, stars with high costs are also gravitated to the black hole. A detailed and fully explained description of this procedure is discussed in section 3.2. At the end of each iteration, the mutation and crossover operators are also applied to the stars in order to improve the performance.

3.1. Creating initial stars

As it is conventional in evolutionary algorithms, the whole information of each particle is stored in an array. This array is known as “particle position”, “chromosome”, and “habitat” in PSO [20,21], GA [22], and CS [23] algorithms, respectively. Here, it is known as: “star position”. The star position is a $1 \times N_{var}$ matrix in solving an N_{var} -dimensional problem. This matrix is defined as Eq. (4):

$$star\ locality = [x_1, x_2, \dots, x_{Nvar}] \quad (4)$$

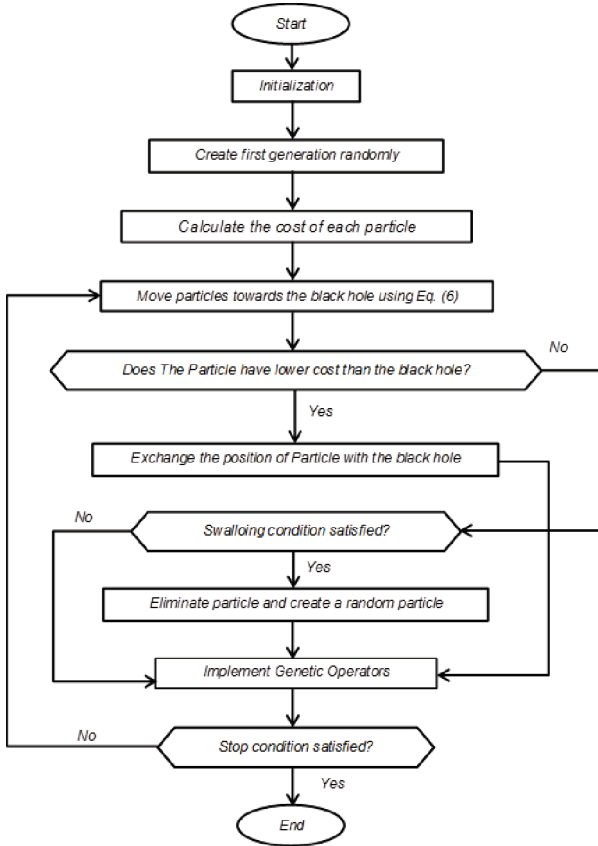


Fig. 1. Flowchart of Modified Black Hole Algorithm.

All elements of this matrix are of floating point digit type. The cost of the point in space is obtained by applying the cost function over the point. This function is defined as:

$$\text{cost} = f(x_1, x_2, \dots, x_{N_{\text{var}}}) \quad (5)$$

At the first step of the optimization, N_{var} stars are generated and distributed in the space randomly. The star with the most desirable cost is considered to be the black hole.

3.2. Moving stars towards the black hole

Following the aforementioned steps, the algorithm should make the stars move towards the black hole. This movement can be formulated as follows:

$$x_{\text{drov}}^{i+1} = x_{\text{drov}}^i + C \times d \quad (6)$$

where x_{drov}^{i+1} and x_{drov}^i are the star location in the $i+1$ th and i th iteration, respectively. C stands for a $1 \times N_{\text{var}}$ matrix. Unlike the standard black hole algorithm,

all the elements of C are uniformly distributed random numbers between 0 and 2. Therefore, stars will search the area around the black hole on both sides. d is the connectivity vector between the particle and the black hole. If a $1 \times N_{\text{var}}$ matrix is chosen as the coefficient of d , the particles will not move along the d vector. Using a vector instead of a number, each component change in a different gain. So, instead of moving across d vector (distance vector), new star moves along a vector in the 2D (two dimensional) plane between star and the black hole. Assume that in a two-dimensional problem, distance vector between the star and the black hole is equal to $d = (a, b)$. If two components of a distance vector are multiplied by a random number between 0 and 2, new vector will certainly be along the vector d . But, if each component is multiplied by a separate random number, the vector may not be along d . The possible vectors are generated in the range of vectors $(2a, 0)$ and $(0, 2b)$. The two vectors $(2a, 0)$ and $(0, 2b)$ are orthogonal and can be considered in the range of $\pm \pi/4$. The particle moves and searches the space in an angular direction with a restricted angle between $-\pi/4$ and $+\pi/4$. This particular characteristic of the proposed algorithm leads the searching phase to be performed with better diversity. Because of emplacement of the random numbers between 0 and 2, the algorithm is able to scour the area around the black hole on both sides for finding the best probable optimum. This procedure is shown in Fig. 2.

After propelling stars towards the black hole, location of a star and the black hole are swapped, only if the assessed particle cost gets better than the previously selected black hole. In other words, particle is named as the new black hole and the previous black hole is known as the particle. Also, the other particles start to move towards the new black hole as shown in Fig. 3.

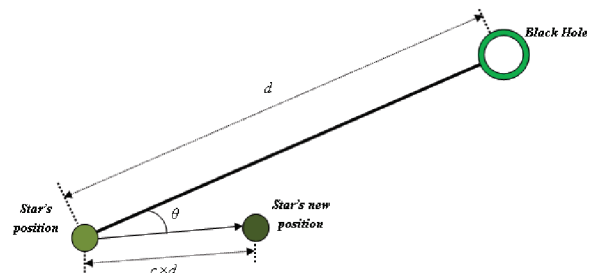


Fig. 2. Moving particles (stars) towards the black hole.

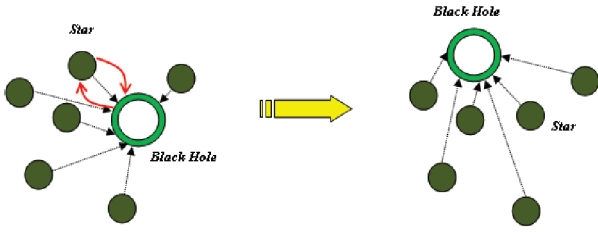


Fig. 3. Exchanging position of a star and black hole when the star presents a more suitable cost.

3.3. Stars swallowing

In a real black hole phenomenon, when the particles reach the particular distance to black hole, they are gravitated and swallowed. The proposed method is also inspired by this procedure, in which, if a particle exceeds the minimum distance to the black hole, it will be swallowed and a new particle will be generated in the search space randomly. This distance can be defined as in Eq. (7) :

$$r = \left(\frac{f_c}{\sum_{n=1}^{N_{pop}} f_n} \right)^2 \quad (7)$$

where f_c stands for the cost of black hole, N_{pop} presents the number of members in each iteration, and f_n is the n th particle cost. Swallowing distance always is a very small number. In this proposed algorithm, the swallow distance of the standard BH algorithm has been squared to get an even smaller radius. Smaller radius is useful when there are a lot of local minimums (or maximums) near the global minimum (or maximum). Therefore, many possible favorable solutions near the black hole are not neglected. This particular specification is one of the remarkable superiorities of MBH algorithm in order to suppress the common drawbacks of getting trapped in a local minimum.

3.4. Limiting stars position

In some cases, stars are placed in a position where the movement towards black hole leads a particular star to be egressed from the search space. As a result, undesirable solutions may be generated. Hence, Eq. (8) is employed to restrict the particle probable positions in the search space frame.

$$x_i = \min\{\max(x_i, LB), UB\} \quad (8)$$

where, x_i is the particle position, LB and UB are the lower and upper bounds of the variable, respectively.

3.5. Implementing mutation and crossover operators

The mutation and crossover operators are applied in order to meliorate the algorithm performance, if there is no improvement in results after 10% of total continual iterations. The first step of genetic operators section is parents selection in which various strategies are available for different kinds of genetic algorithms [24]. Here, a random approach for parents selection has been used to improve CPU time of the presented algorithm.

Crossover is one of the important operators in the well-known evolutionary algorithms such as GA and DE [25]. Crossover probability should be defined to perform the crossover operation, where its value is planned to be 70% in the MBH algorithm. Crossover is mainly splitted into three groups: single-point crossover, double-point crossover, and uniform crossover. The proposed algorithm employs uniform crossover due to its high level of diversity. The general form of a uniform crossover between two particles z_1 and z_2 with the output offsprings y_1 and y_2 , is as follows:

$$y_1 = \alpha z_1 + (1 - \alpha) z_2 \quad (9)$$

$$y_2 = \alpha z_2 + (1 - \alpha) z_1 \quad (10)$$

where α is a $1 \times N_{var}$ vector consisting of real numbers limited in $[-\Delta, \Delta+1]$ range, and Δ is a number about 0.1. Coefficient α in Eqs. (9) and (10) is a number between 0 and 1 like as in the standard genetic algorithm. With this range of α , offsprings are generated between parents. But, it may be available vantage points around parents (not between them). Therefore, a bigger range for coefficient α is considered as $[-\Delta, \Delta+1]$. Δ should be chosen such that the majority of children between parents and few outside the interval. That is why the value of 0.1 is reasonable. So, offsprings position are not limited by their parents. The bigger Δ causes a wider range out of parents position which is not desirable.

Another genetic operators used in the proposed method is mutation [26]. First, the j th entry of the z_1 parent is selected and the mutated offspring is generated according to the following equations:

$$y_1 = z_1 \quad (11)$$

```

Begin
  Step1: Initializations. Initialize first Generation randomly and save each stars position in Star position array (DL);
  Step2: Evaluate each star's cost (C) and determine black hole;
  Step3: while the stop criteria is not satisfied do
    for i=1:Npop
      Move i th star towards the black hole;
      flag: Evaluate ith stars cost (Ci);
      if (Ci < Cblack hole)
        Replace positions of i th star and black hole;
      end if
      Calculate swallowing radius (r);
      if (DLi - DLblack hole) < r
        i th star is swallowed and a random star will be generated instead;
      end if
    end flag
  end for
  Implement Genetic operators;
  Go to flag;

  Step4: end while
End.

```

Fig. 4. Pseudo-code for Modified Black Hole Algorithm.

$$s = \frac{UB-LB}{10} \quad (12)$$

$$y_1(j) = x_1(j) + s * randn \quad (13)$$

where $z_i(j)$ is the j th entry of locality array of a parent star. $y_i(j)$ is the j th entry of locality array of a offspring star. $randn$ is a normally distributed random number between 0 and 1. The “mutation probability” is considered to be 20% in the present study, which means the 20% of the stars will be mutated. It should be noted that the generated offsprings should be in the defined bounds.

Eventually, initial population and generated offsprings are sorted in terms of the cost, and N_{pop} stars with the better cost introduced as the new generation.

The pseudo code of Modified Black Hole algorithm has been shown in Fig. 4.

4. Experimental Results

In this section, performance of the MBH algorithm is evaluated. The test of reliability, efficiency and validation of optimization algorithms is frequently carried out using a set of standard benchmarks or test functions [27]. The benchmark problems are categorized to low and high dimensional problems. The low dimensional ones are used to test the speed of convergence, however the high dimensional ones are

good criterions to evaluate the computing power of an algorithm. In this work, the MBH algorithm has been applied to 20 different benchmark problems including ten low and ten high dimensional problems. Data sets of the problems are shown in Table 1.

The famous peak problem has been selected in order to observe the objective behavior of the MBH algorithm. The contour of the peak problem is shown in Fig. 5 and the corresponding equation is as follows:

$$z = xe^{-(x^2+y^2)} \quad , \quad -2 < x, y < 2 \quad (14)$$

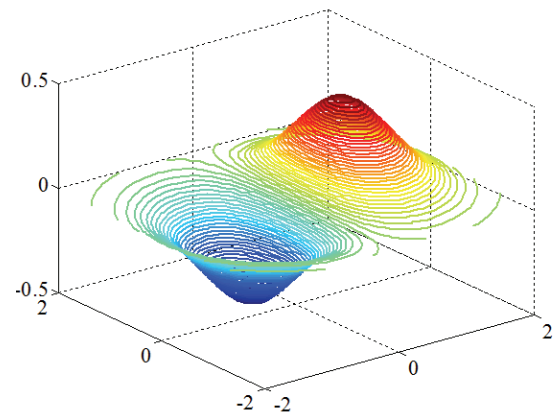


Fig. 5. 3-D Contour plot of Peak function.

Table 1. Benchmark Problems.

	Name	Function	Dimensions	Variables Bound	Global Minimum
<i>High-Dimension Benchmark Functions</i>					
P1	Alpine	$f(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	20	$X \in [-10,10]$	0
P2	Alpine 2	$f(x) = \prod_{i=1}^D \sqrt{x_i} \sin(x_i) $	20	$X \in [-10,10]$	0
P3	csendes	$f(x) = \sum_{i=1}^D x_i^6 \left(2 + \sin\left(\frac{1}{x_i}\right)\right)$	20	$X \in [-1,1]$	0
P4	Deb1	$f(x) = -\frac{1}{D} \sum_{i=1}^D \sin^6(5\pi x_i)$	20	$X \in [-1,1]$	-1
P5	Quintic	$f(x) = \sum_{i=1}^D x_i^5 + 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4 $	20	$X \in [-10,10]$	0
P6	Rastrigin	$f(x) = 10D \sum_{i=1}^D x_i^2 - 10 \cos(2\pi x_i)$	20	$X \in [-5.12, 5.12]$	0
P7	Salomon	$f(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$	20	$X \in [-100,100]$	0
P8	Schwefel2.21	$f(x) = \max\{ x_i , 1 \leq i \leq D\}$	20	$X \in [-100,100]$	0
P9	Schwefel2.26	$f(x) = -\frac{1}{D} \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$	20	$X \in [-500,500]$	-418.983
P10	Sphere	$f(x) = \sum_{i=1}^D x_i^2$	20	$X \in [-10,10]$	0
<i>Low-Dimension Benchmark Functions</i>					
P11	Ackley 2	$f(x) = -200e^{-0.02\sqrt{\frac{x_1^2 + x_2^2}{2}}}$	2	$X \in [-32,32]$	-200
P12	Bartels Conn	$f(x) = x_1^2 + x_2^2 + x_1x_2 + \sin(x_1) + \cos(x_2) $	2	$X \in [-500,500]$	1
P13	Parsopoulos	$f(x) = \cos^2(x_1) + \sin^2(x_2)$	2	$X \in [-5,5]$	0
P14	Periodic	$f(x) = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1e^{-(x_1^2 + x_2^2)}$	2	$X \in [-10,10]$	0.9
P15	Price	$f(x) = (x_1 - 5)^2 + (x_2 - 5)^2$	2	$X \in [-500,500]$	0
P16	Rotated Ellipse	$f(x) = 7x_1^2 - 6\sqrt{3}x_1x_2 + 13x_2^2$	2	$X \in [-500,500]$	0
P17	Scahffer	$f(x) = 0.5 + \frac{\sin^2((x_1^2 + x_2^2)^2) - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}$	2	$X \in [-500,500]$	0
P18	Table 1	$f(x) = \cos(x_1)\cos(x_2)e^{\frac{1 - \sqrt{x_1^2 + x_2^2}}{\pi}} $	2	$X \in [-10,10]$	0
P19	Ursem 4	$f(x) = 3\sin(0.5\pi x_1 + 0.5\pi) \times \frac{2 - \sqrt{x_1^2 + x_2^2}}{4} $	2	$X \in [-2,2]$	0
P20	Hartman 1	$f(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^3 a_{ij}(x_i - p_{ij})^2]$, Parameters a, c, p are presented in Table 2.	3	$X \in [0,1]$	-3.8628

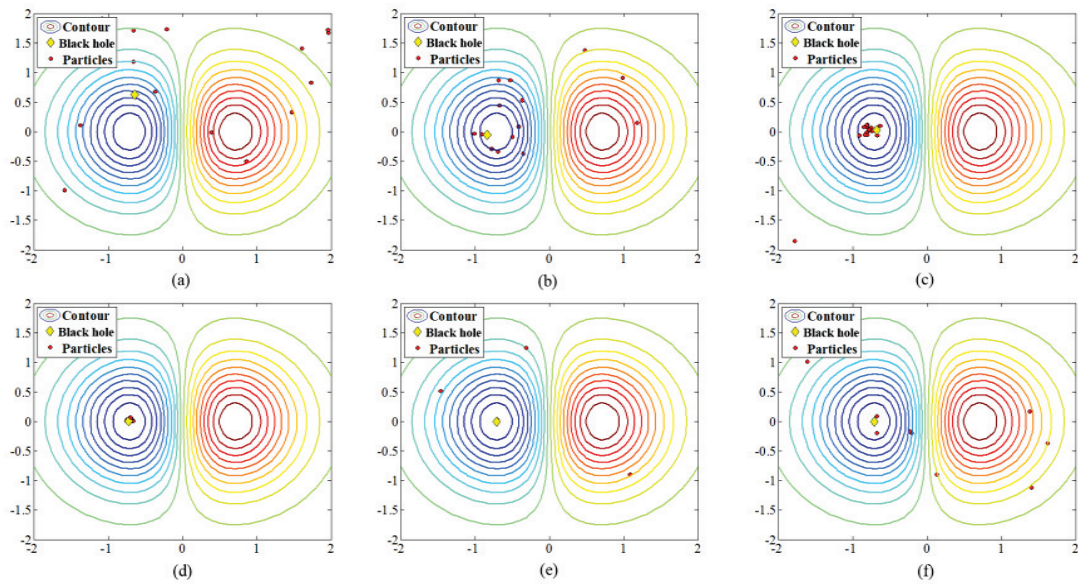


Fig. 6. Positions of stars and black hole in (a) Initial generation, (b) 1st iteration, (c) 2nd iteration, (d) 3rd iteration, (e) 4th iteration and (f) 5th iteration.

Starting the problem with an initial population of 15 particles. The algorithm behavior from the initial generation to the fourth iteration has been presented in Fig. 6. Initial population of the random generated particles is shown in Fig. 6(a). Location of the particles and the black hole after the first iteration is also presented in Fig. 6(b). As shown, location of the black hole has been changed from $(-0.64, 0.62)$ with -0.2876 of cost to the point $(-0.83, -0.05)$ with -0.4141 of cost. In the third iteration, all the particles (stars) have been gathered around the minimum, however, they have not yet reached the global minimum. It should be noted that, in the fourth iteration, some of stars have been distanced from the black hole. It indicates the point in which these stars are too close to the black hole. The black hole starts to omit them and locate these stars in a random point. The algorithm is reached to the global minimum coordinates $(-0.708, 0.002)$ with cost of -0.4289 in the fifth iteration of the test. It is obvious that more particles have been omitted in this stage due to their close distance to the black hole. Efficiency and the power of proposed method in terms of detecting the global minimum is expressed in the peak problem.

Now, a two dimensional problem with more than one local minima is considered to test the MBH algorithm performance. A two dimensional problem has been presented in Eq. (15):

$$f(x, y) = x \sin(4x) + 1.1y \sin(2y) \quad , \quad 0 < x, y < 10 \quad (15)$$

There are 18 minimum points in the above-mentioned range, where the point $(9.0390, 8.6681)$ with the cost value of -18.5547 is the global minimum. The three-dimensional overview of the function is presented in Fig. 7.

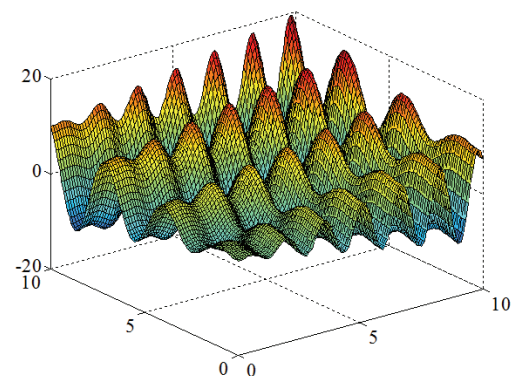


Fig. 7. 3-D plot of $f(x,y)$ function.

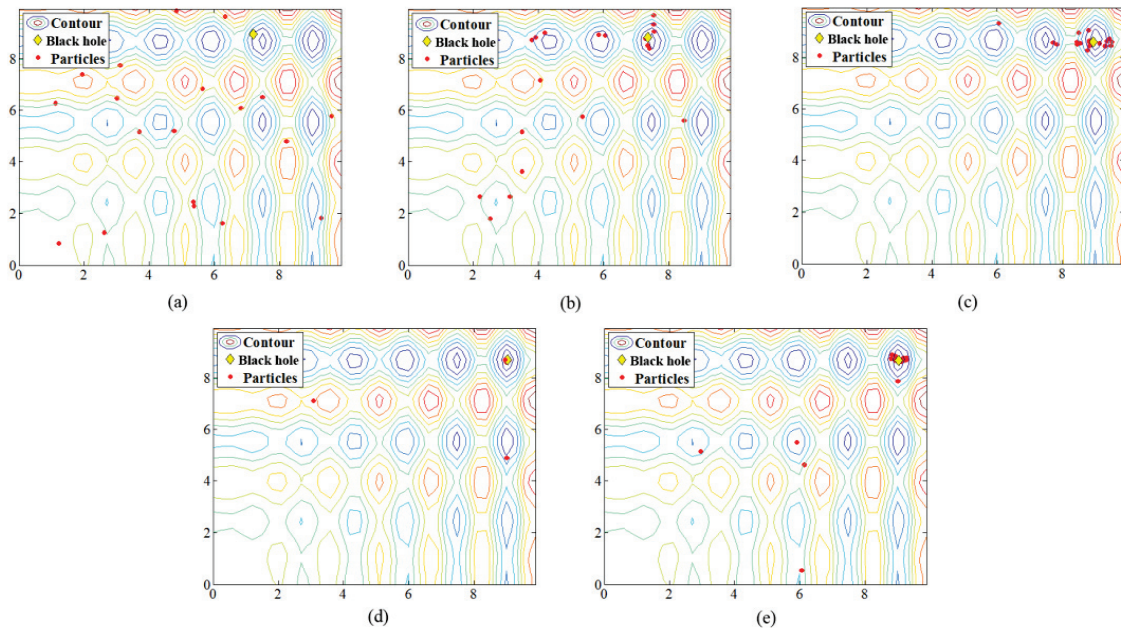


Fig. 8. Positions of stars and black hole in (a) Initial generation, (b) 1st iteration, (c) 3rd iteration, (d) 5th iteration and (e) 10th iteration.

Starting with the initial population of 20, the progress procedure of solution after ten iterations is shown in Fig. 8. In the first iteration, the black hole is placed in local minima. It will gradually find its way near the desired global minimum up to the third iteration. In this stage, the black hole is at (8.9543, 8.5847) with the cost of -17.9116. Its distance is reduced to the global minimum inchmeal, and in the seventh iteration it will arrive to the best point (9.0387, 8.6689). Comparing Fig. 8(e). and Fig. 8(d)., it is clear that the omitting section of the very high particles has been done

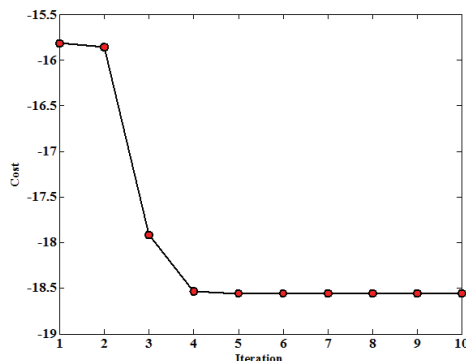


Fig. 9. Cost minimization for the problem $f(x,y)$.

successfully, and the particles contiguous to the black hole have been generated in different coordinates. The diagram of the cost reduction versus iteration is shown in Fig. 9 indicating the point in the fourth iteration. The algorithm has been reduced its distance to the desired minimum and finally, it reaches to its best cost at the seventh iteration.

As cited earlier in this section, 20 benchmark problems have been tested, and results are obtained. In addition, comparisons with several well-known methods such as PSO, BA, BH, HS [27], and FA, has been presented in order to analyze the performance of the MBH algorithm for different problems. Table 2 shows the parameters of each optimization algorithms. There are various parts with random behavior in these types of algorithms. If such algorithms are applied to different problems once, it may not express exact strengths and weaknesses of experimented method. Therefore, multiplicity of tests should be increased to avoid the lack of cogent gauge and the random behavior effect. For this purpose, procedure of algorithms have been repeated 50 times for each problem to validate the results and obtain clear outcomes. The obtained solutions are presented in Table 3 and Table 4 considering the four factors, namely best,

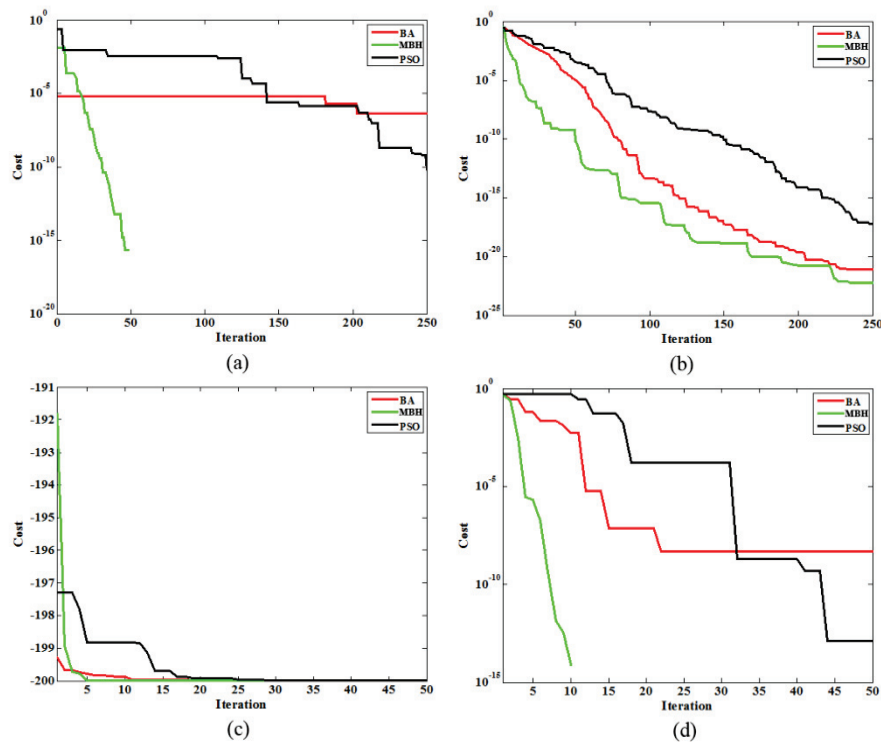


Fig. 10. Comparison of the cost convergence of BA, MBH and PSO for (a) P1, (b) P3, (c) P11 and (d) P17 problems.

Table 2. Algorithms parameters.

Algorithm	Parameters
PSO	Personal Learning Coefficient (c1) and Global Learning Coefficient (c2): are determined by Constriction Coefficients method [21].
BA	Number of Scout Bees: 30 Number of Selected Sites: 15 Number of Selected Elite Sites: 6 Number of Recruited Bees for Selected Sites: 15 Number of Recruited Bees for Elite Sites: 12 Neighborhood Radius Damp Rate: 0.99
ICA	Number of Empires/Imperialists: 10 Selection Pressure: 1 Assimilation Coefficient: 2 Revolution Probability: 0.1 Revolution Rate: 0.05 Colonies Mean Cost Coefficient: 0.1
FA	Light Absorption Coefficient: 1 Attraction Coefficient Base Value: 2 Mutation Coefficient: 0.2 Mutation Coefficient Damping Ratio: 0.99

worst, average, and standard deviation. Each algorithm has been initialized with 50 trials. The crossover and mutation operators are utilized and adopted to be 70% and 20%, respectively. The high dimensional problems are presented in Table 3. Since there are many decision variables available in problems P1 to P10, the number of iterations in each run has been set to 250. Therefore, enough time is provided for algorithm to seek the search space completely, however the number of iterations for the low dimensional problems has been considered to be 50. Table 4 shows the resultant of applying various algorithms to low dimensional sets P11 to P20. The modified black hole is marked as the best solutions available against its counterparts, showing the fastest convergence ratio to reach the desirable minimum.

In order to improve the comparison procedure, the cost versus iteration diagram of PSO, MBH, and BA for a random test has been plotted in Fig. 10. The Alpine problem (P1) is studied in Fig. 10(a). The MBH algorithm converges to the desired global minimum with the best cost in the 49th iteration, while the two

Table 3. Constants of Hartman1 problem.

I	C_i	a_{ij}			p_{ij}	
		J=1	2	3	J=1	2
1	1	3	10	30	0.3689	0.117
2	1.2	0.1	10	35	0.4699	0.4387
3	3	3	10	30	0.1091	0.8732
4	3.2	0.1	10	35	0.03815	0.5743

other methods do not reach the minimum after 250th iteration. Fig. 10(b) presents solutions of the discussed algorithms to a high dimensional problem called as Csendes (P3). Although all the three algorithms start the process almost alike, nearly with the same cost, MBH presents the best speed of convergence and better performance compared with the two other algorithms after 250 iteration. The cost minimization plot of the three aforementioned algorithms for the problems called as Ackley 2 (P11) and Schaffer (P17) is shown in Fig. 10(c)-(d). As it is clear in Fig. 10(c), although the initial cost of MBH is much more than the others, but the best global optimum is obtained in the fifth iteration whereas the iteration number for the best solution is 11 for BA and 25 for PSO. Another example is shown in Fig. 10(d) where the seeking process is started with approximately the same initial cost. MBH has reached the best optimum in the eleventh iteration, while BA and PSO are incapable of reaching the optimum even after 50 times of iterations.

Finally, the worst, mean, and best run of the cost minimization plot of MBH algorithm for six different problems are presented in Fig. 11. As shown in Fig. 11(a) and Fig. 11(f), when the MBH algorithm reach to the 100% of accuracy i.e. zero for the cited problems, the figures are corrupted. The reason is that the logarithmic plots are incapable of representing zero.

5. Conclusion

In this paper, the modified Black Hole algorithm has been proposed for solving optimization problems. A new approach is employed to overcome the probable drawbacks caused by the trapping effect, where trapping is suppressed and particles close to the black hole are swallowed with a new swallowing range if they exceed the minimum distance to the black hole. Afterwards, a new particle is generated in the search space randomly to avoid probable disturbances of trapping effect. The proposed algorithm has been applied to 20 different benchmark problems on 50 tests, and the results are compared with the standard Black Hole algorithm and its other counterparts. It is clear that the proposed method presents various remarkable superiorities than the other well-known optimization algorithms such as BH, PSO, BA, ICA, and FA. Also, the MBH outperforms the others in 17 of 20 problems. It is more prominent that the exact solutions (100% accuracy in the best runs) are obtained for 10 problems, showing the power and highly accurate performance of the Modified Black Hole algorithm.

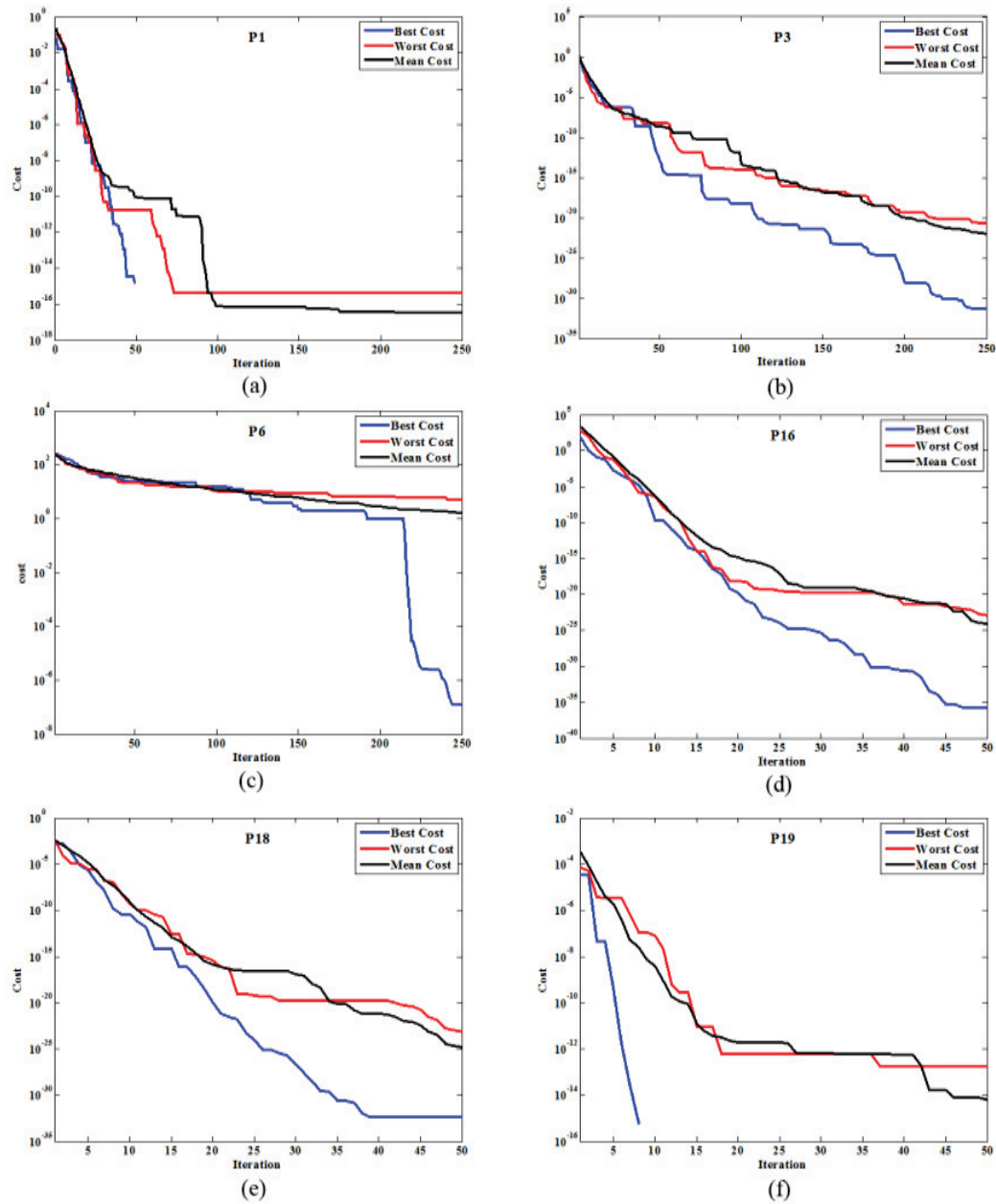


Fig. 11. Cost minimization of Modified Black Hole algorithm in best, worst and mean test in 50 tests for (a) P1, (b) P3, (c) P6, (d) P16, (e) P18 and (f) P19 problems, where the exact solutions have been obtained in figures (a) and (f) in the best run.

Table 4. Best, worst, mean and the standard deviation of available solution in MBH and BH, PSO, BA, ICA, FA for low-dimension benchmark problems.

	MBH	BH	PSO	BA	ICA	FA	
P11	-200	-200	-199.9999	-199.9996	-200	-199.9995	Best
	-199.9999	-199.9968	-199.9748	-199.9879	-199.9998	-199.9795	Worst
	-200	-199.9991	-199.9937	-199.9955	-200	-199.9921	Mean
	1.6986e-005	0.0007738	0.0050905	0.0029244	3.304e-005	0.0048177	Std.
P12	1	1	1.00005	1.0004	1	1.0003	Best
	1	1.033908	1.03204	1.0312	1	1.0307	Worst
	1	1.003038	1.00819	1.0074	1	1.0082	Mean
	1.2781e-010	0.005630	0.007895799	0.0061017	4.1683e-006	0.00693	Std.
P13	7.4988e-033	1.536e-018	1.2212e-010	1.8528e-010	8.7737e-020	1.2567e-009	Best
	1.148e-023	0.013345	0.0009338	3.6351e-007	1.9311e-011	4.721e-007	Worst
	2.7415e-025	0.0014562	2.5437e-005	4.8615e-008	6.5184e-013	1.1754e-007	Mean
	1.6321e-024	0.0028489	0.0001323	6.7474e-008	2.7816e-012	1.0451e-007	Std.
P14	0.9	0.9	0.9	0.9	0.9	0.9	Best
	1	1.0136	1.0003	1	1	1	Worst
	0.9164	0.93867	0.94773	0.92014	0.93921	0.992	Mean
	0.036639	0.049791	0.050002	0.04034	0.048462	0.037405	Std.
P15	0	1.9824e-021	3.4774e-006	2.6024e-006	9.1726e-015	4.5209e-005	Best
	1.9474e-021	6.4145e-007	0.019067	0.00056819	1.0025e-005	0.0042154	Worst
	3.8956e-023	2.5181e-008	0.0031429	0.00012437	2.1503e-007	0.0010889	Mean
	2.7541e-022	9.5795e-008	0.0047193	0.00013662	1.4173e-006	0.00093021	Std.
P16	1.7809e-036	4.1122e-019	9.8908e-006	0.00010058	6.4237e-013	9.0579e-005	Best
	1.1952e-023	0.027406	0.047892	0.041529	0.00032499	1.9692	Worst
	1.0299e-024	0.0014927	0.0077412	0.01229	1.2746e-005	0.051312	Mean
	2.6402e-024	0.0053704	0.010168	0.011859	4.8577e-005	0.27717	Std.
P17	0	0	1.1102e-016	2.8522e-013	0	2.3981e-014	Best
	0	2.3804e-007	5.3775e-009	1.4302e-008	0.0092577	9.9654e-009	Worst
	0	4.7607e-009	1.5558e-010	7.388e-010	0.0003093	1.3847e-009	Mean
	0	3.3664e-008	7.7712e-010	2.2204e-009	0.0015614	2.6762e-009	Std.
P18	4.7945e-033	3.1004e-025	3.8976e-011	3.2972e-011	1.1365e-020	4.3332e-011	Best
	6.9868e-024	2.6222e-011	0.00017361	3.0583e-007	2.9605e-013	3.2899e-007	Worst
	1.511e-025	5.4595e-013	2.1481e-005	3.4525e-008	9.1173e-015	7.2566e-008	Mean
	9.8815e-025	3.7084e-012	3.8908e-005	5.1877e-008	4.2239e-014	9.0057e-008	Std.
P19	0	0	1.452e-010	5.7192e-011	0	0	Best
	1.8171e-013	1.5377e-007	2.1603e-005	2.6426e-008	3.2401e-013	8.1493e-008	Worst
	6.5718e-015	5.7845e-009	1.5408e-006	2.6765e-009	1.0084e-014	2.7983e-009	Mean
	2.917e-014	2.8146e-008	4.4988e-006	5.3208e-009	4.7946e-014	1.1505e-008	Std.
P20	-3.8628	-3.8626	-3.8628	-3.8628	-3.8628	-3.8628	Best
	-3.8628	-3.8259	-3.8549	-3.8628	-3.8628	-3.8628	Worst
	-3.8628	-3.85	-3.8626	-3.8628	-3.8628	-3.8628	Mean
	1.2449e-008	0.0077845	0.0011145	1.2817e-007	5.9477e-008	4.9129e-007	Std.

Table 5. Best, worst, mean and the standard deviation of available solution in MBH and BH, PSO, BA, ICA, FA for low-dimension benchmark problems.

	MBH	BH	PSO	BA	ICA	FA	
P11	-200	-200	-199.9999	-199.9996	-200	-199.9995	Best
	-199.9999	-199.9968	-199.9748	-199.9879	-199.9998	-199.9795	Worst
	-200	-199.9991	-199.9937	-199.9955	-200	-199.9921	Mean
	1.6986e-005	0.0007738	0.0050905	0.0029244	3.304e-005	0.0048177	Std.
P12	1	1	1.00005	1.0004	1	1.0003	Best
	1	1.033908	1.03204	1.0312	1	1.0307	Worst
	1	1.003038	1.00819	1.0074	1	1.0082	Mean
	1.2781e-010	0.005630	0.007895799	0.0061017	4.1683e-006	0.00693	Std.
P13	7.4988e-033	1.536e-018	1.2212e-010	1.8528e-010	8.7737e-020	1.2567e-009	Best
	1.148e-023	0.013345	0.0009338	3.6351e-007	1.9311e-011	4.721e-007	Worst
	2.7415e-025	0.0014562	2.5437e-005	4.8615e-008	6.5184e-013	1.1754e-007	Mean
	1.6321e-024	0.0028489	0.0001323	6.7474e-008	2.7816e-012	1.0451e-007	Std.
P14	0.9	0.9	0.9	0.9	0.9	0.9	Best
	1	1.0136	1.0003	1	1	1	Worst
	0.9164	0.93867	0.94773	0.92014	0.93921	0.992	Mean
	0.036639	0.049791	0.050002	0.04034	0.048462	0.037405	Std.
P15	0	1.9824e-021	3.4774e-006	2.6024e-006	9.1726e-015	4.5209e-005	Best
	1.9474e-021	6.4145e-007	0.019067	0.00056819	1.0025e-005	0.0042154	Worst
	3.8956e-023	2.5181e-008	0.0031429	0.00012437	2.1503e-007	0.0010889	Mean
	2.7541e-022	9.5795e-008	0.0047193	0.00013662	1.4173e-006	0.00093021	Std.
P16	1.7809e-036	4.1122e-019	9.8908e-006	0.00010058	6.4237e-013	9.0579e-005	Best
	1.1952e-023	0.027406	0.047892	0.041529	0.00032499	1.9692	Worst
	1.0299e-024	0.0014927	0.0077412	0.01229	1.2746e-005	0.051312	Mean
	2.6402e-024	0.0053704	0.010168	0.011859	4.8577e-005	0.27717	Std.
P17	0	0	1.1102e-016	2.8522e-013	0	2.3981e-014	Best
	0	2.3804e-007	5.3775e-009	1.4302e-008	0.0092577	9.9654e-009	Worst
	0	4.7607e-009	1.5558e-010	7.388e-010	0.0003093	1.3847e-009	Mean
	0	3.3664e-008	7.7712e-010	2.2204e-009	0.0015614	2.6762e-009	Std.
P18	4.7945e-033	3.1004e-025	3.8976e-011	3.2972e-011	1.1365e-020	4.3332e-011	Best
	6.9868e-024	2.6222e-011	0.00017361	3.0583e-007	2.9605e-013	3.2899e-007	Worst
	1.511e-025	5.4595e-013	2.1481e-005	3.4525e-008	9.1173e-015	7.2566e-008	Mean
	9.8815e-025	3.7084e-012	3.8908e-005	5.1877e-008	4.2239e-014	9.0057e-008	Std.
P19	0	0	1.452e-010	5.7192e-011	0	0	Best
	1.8171e-013	1.5377e-007	2.1603e-005	2.6426e-008	3.2401e-013	8.1493e-008	Worst
	6.5718e-015	5.7845e-009	1.5408e-006	2.6765e-009	1.0084e-014	2.7983e-009	Mean
	2.917e-014	2.8146e-008	4.4988e-006	5.3208e-009	4.7946e-014	1.1505e-008	Std.
P20	-3.8628	-3.8626	-3.8628	-3.8628	-3.8628	-3.8628	Best
	-3.8628	-3.8259	-3.8549	-3.8628	-3.8628	-3.8628	Worst
	-3.8628	-3.85	-3.8626	-3.8628	-3.8628	-3.8628	Mean
	1.2449e-008	0.0077845	0.0011145	1.2817e-007	5.9477e-008	4.9129e-007	Std.

References

1. P. Agrawal, S. Mehta, Nature-inspired algorithms: state-of-art, problems and prospects, *Int. J. Comp. App.*, **100** (14) (2014) 14-21.
2. S. Cholavendhan, R. Siva Kumar, M. Karnan, A survey on application of bio-inspired algorithms, *Int. J. Comp. Sci. Inform. Tech.*, **5** (1) (2014) 366-370.
3. J. Zhang, K. Liu, Y. Tan, and X. He, Random Black Hole Particle Swarm Optimization and Its Application, in *Proc. IEEE Int. Conf. Neural Networks & Signal Processing*, 2008, pp. 539-365.
4. A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Inform. Sciences*, **222** (2013) 175-184.
5. M. Nemati, H. Momeni, N. Bazrkar, Binary Black Holes Algorithm, *Int. J. Comp. App.*, **29** (6) (2013) 36-42.
6. H.R.E.H. Boucekara, Optimal power flow using black-hole-based-optimization approach, *App. Soft Comput.*, **24** (2014) 879-888.
7. R. Azizpanah-Abarghoee, T. Niknam, F. Bavafa, M. Zare, Short-term Scheduling of thermal power systems using gradient based modified teaching-learning optimizer with black hole algorithm, *Electr. Pow. Sys. Res.*, **108** (2014) 16-34.
8. M. Nemati, R. Salimi, N. Bazrkar, Black Holes Algorithm: A swarm algorithm inspired of black holes for optimization problems, *Int. J. Artif. Int.*, **2** (3) (2013) 143-150.
9. M. Nemati, H. Momeni, Black holes algorithm with fuzzy Hawking radiation, *Int. J. Sci. Tech. Res.*, **3** (6) (2014) 85-88.
10. H.R.E.H. Boucekara, Optimal design of electromagnetic devices using a black-hole-based optimization technique, *IEEE T. Magn.*, **49** (2013) 5709-5714.
11. M. Doraghinezhad, H. Nezamabadi-pour, Black Hole: a new operator for gravitational search algorithm, *Int. J. Comput. Int. Sys.*, **7** (5) (2014) 1-18.
12. M. Eskandarzadehahamdamy, B. Masoumi, O. Sojodishijani, A new hybrid algorithm based on black hole optimization and bisecting k-means for cluster analysis, in *Proc. 22nd Iranian Conf. Electrical Engineering*, 2014, 1075-1079.
13. S. Yaghoobi, S. Hemayat, H. Mojallali, Image Grey-level enhancement using black hole algorithm, in *Proc. 2nd Int. Conf. Pattern Recognition and Image Analysis*, 2015, pp. 1-5.
14. E. Pashaei, M. Ozen, N. Aydin, An application of black hole algorithm and decision tree for medical problem, in *Proc. 15nd In. Conf. Bioinformetica and Bioengineering*, 2015, 1-6.
15. K. Premalatha, R. Balamurugan, A nature inspired swarm based stellar-mass black hole for engineering optimization, in *Proc. Int. Conf. Electrical, Computer and Communication Technologies*, 2015, 1-8.
16. K. Lenin B. Ravindhranath Reddy, M. Surya Kalavathi, Dwindling of active power loss by enhanced black hole algorithm, *Int. J. Res. Electron. Comm. Tech.*, **1** (4) (2014) 11-15.
17. J. Liu, J. Luo, Optimal economic emission hydrothermal scheduling based on black hole theory and annual profit analysis considering FGD, in *Proc. IET Conf. Renewable Power Generation*, 2014, 1-6.
18. D. Rodrigues, C.C.O. Ramos, J.P. Papa, Black hole algorithm for non-technical losses characterization, in *Proc. 6th Latin American Symp. Circuits & Systems*, 2015, 1-4.
19. R. Wald, *General Relativity* (Chicago: Univ. of Chicago Press, pp. 299-300, 1984)
20. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proc. IEEE Int. Con. Neural Networks*, vol. 4, 1995, pp. 1942- 1948.
21. Y. Shen, Wang, C. Tap, Particle Swarm Optimization with Novel Processing Strategy and its Application, *Int. J. Comput. Int. Sys.*, **4** (1) (2011) 100-111.
22. C. Moon, J. Kim, G. Choi, Y. Seo, An efficient genetic algorithm for the traveling salesman problem with precedence constraints, *Eur. J. Oper. Res.*, **140** (3) (2002) 606-617.
23. R. Rajabioun, Cuckoo Optimization Algorithm, *Appl. Soft Comput.*, **11** (8) (2011) 5508-5518.
24. N. M. Razali, J. Geraghty, Genetic Algorithm Performance with Different Selection Strategies in Solving TSP, in *Proc. World Congress on Engineering*, 2011, pp. 156-162.
25. R. Storn, K. Price, Differential Evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, *J. Glob. Optim.*, **11** (4) (1997) 341-359.
26. X.-S. Yang, Harmony Search as a Metaheuristic Algorithm, in *Music-Inspired Harmony Search Algorithm: Theory and Applications* (Springer, Berlin, 2009) pp. 1-14.
27. Momin Jamil and Xin-She Yang, A literature survey of benchmark functions for global optimization problems, *Int. J. Math. Model. Numer. Optimizat.*, **4** (2) (2013) 150-194.