A NEW HYBRID GRAVITATIONAL SEARCH – BLACK HOLE ALGORITHM

Suad Khairi Mohammed
Faculty of Electrical and Electronics Engineering
University Malaysia Pahang
Pahang, Malaysia
suad.khairim@gmail.com

Zuwairie Ibrahim
Faculty of Electrical and Electronics Engineering
University Malaysia Pahang
Pahang, Malaysia
zuwairie@ump.edu.my

Hamdan Daniyal
Faculty of Electrical and Electronics Engineering
University Malaysia Pahang
Pahang, Malaysia
hamdan@ump.edu.my

Nor Azlina Ab. Aziz
Faculty of Engineering and Technology
Multimedia University
75450 Melaka, Malaysia
azlina.aziz@mmu.edu.my

Abstract—This paper studies the solution of optimization problems using a new hybrid population-based algorithm. The Gravitational Search – Black Hole Algorithm (GSBHA) is proposed as a combination of the Black Hole Algorithm (BHA) and Gravitational Search Algorithm (GSA). The main idea is to improve the standard BHA by using GSA. To evaluate the performance of GSBHA, standard test functions of CEC 2014 for real-parameters are used to compare the hybrid algorithm with both the standard BHA and GSA algorithms in evolving the best solution. The results obtained demonstrate better performance of the hybrid algorithm and better capability to escape from local optimums with faster convergence than the standard BHA and GSA.

Keywords— Optimization, Black Hole, Gravitational Search Algorithm, Hybrid algorithms and CEC14.

1. INTRODUCTION

To solve optimization problems, efficient search or optimization algorithms are needed. There are many optimization algorithms, which can be classified in many ways, depending on the focus and characteristics. If the derivative or gradient of a function is the focus, optimization can be classified as gradient-based algorithms and derivative-free or gradient-free algorithms [1] and [2]. Gradient-based algorithms, such as hill-climbing, use derivative information, and they are often very efficient.

From a different perspective, optimization algorithms can be classified into trajectory-based and population-based [3] and [4]. A trajectory-based algorithm typically uses a single agent or one solution at a time and traces a path as the iterations continue [5] and [6]. Hill-climbing is trajectory-based, and it links the starting point with the final point via a piecewise zigzag path. Another important example is simulated annealing, which is a widely used metaheuristic algorithm. Population-based algorithms, such as particle swarm optimization (PSO), use multiple agents that interact and trace multiple paths (Kennedy and Eberhardt, 1995) [8].

Optimization algorithms can also be classified as deterministic or stochastic. If an algorithm works in a mechanical deterministic manner without any random nature, it is called deterministic. Hill-climbing and downhill simplex are good examples of deterministic algorithms. If there is some randomness in the algorithm, the algorithm usually reaches a different point every time the algorithm is executed, even though the same initial point is used. Genetic algorithms and PSO [11] are good examples of stochastic algorithms [7]. In this case, the algorithms can be divided into local and global search algorithms. Local search algorithms typically converge towards a local optimum, not necessarily (often not) the global optimum, and such an algorithm is often deterministic and has no ability to escape from local optima. Simple hill-climbing is an example [29]. For global optimization, local search algorithms are not suitable, and global search algorithms should be used. Optimization approaches can be classified into exact algorithms and approximate algorithms [21] and [23]. Exact approaches are able to find the exact optimal answer [26] and [27]. Approximate algorithms are able to find good answers (near the optimal solutions) [24]. These algorithms are divided into two groups of heuristic algorithms and meta-heuristic algorithms. Heuristic approaches create suitable and good solutions, which are normally not the best solution. Meta-heuristic approaches have been formed according to inspiration by nature, physics and humans. In recent years, many of these algorithms and their improved forms have been successfully applied to various problems of engineering optimization [8].

Algorithms can be a mixed type or hybrid, which use some combination of deterministic components with randomness or combine one algorithm with another to design more efficient algorithms [15] and [18]. Hybridization [16] and [17] of some of these algorithms is intended to combine the strengths of their respective heuristic techniques into a better algorithm [19] and [22]. This new algorithm should produce solutions closer to the optimal, or in less time, or both [32] and [33]. An algorithm that produces satisfactory results in less time can also be applied to large problems [30]. In general, hybridization can usually make some improvements in terms of either computational speed or accuracy.

In the current literature, the Gravitational Search Algorithm (GSA) and Black Hole algorithm (BHA) are used to obtain suitable solutions and better performance than conventional calculation methods for some complicated and difficult real-world optimization problems. The motivation in this experiment to form a hybrid between these algorithms refers to the combination of the characteristics of (GSA), such as gravity, motion, objects, velocity and masses, and the characteristics of (BHA), such as gravity of a black hole, stars, movement of stars and positions. Although the ability of GSA [7] has been proved by many researches on different problems, there is also the possibility of premature convergence [31]. Because the guiding force attracts the objects to each other, the algorithm performance falls if the objects converge to a non-optimal solution [5],[20] and [25].

The goal of these algorithms is to find the best outcome (global optimum) among all possible inputs. To do this, a heuristic algorithm should be equipped with two major characteristics to ensure finding the global optimum. These two characteristics are exploration and exploitation. For good performance, it is necessary to make a suitable trade-off between exploration and exploitation capabilities [5].

Solving optimization problems based on improved BHA is proposed here. The obtained results are compared with two other optimization algorithms. This paper is organized as follows: in section 2, BHA and its characteristics are reviewed. In section 3, GSA and the basic concept are briefly explained. In section 4, the proposed BHGSA is presented. In section 5. The experimental parameters and the experimental results are presented in section 6. Finally, the section 7 presents the conclusions of the experiments.

2. BLACK HOLE ALGORITHM

The Black Hole Algorithm (BHA) is a recent population-based metaheuristic algorithm inspired by the physical phenomenon of black holes. A black hole is a region of space packed with so much matter that its own gravity prevents anything from escaping – even a ray of light. Black holes can form when massive stars run out of fuel and collapse under their own weight, creating such strong gravity that they disappear from view. BHA is population-based and is one of the newest approaches that has been created and used successfully for solving optimization problems. The population of candidate solutions (stars) in this algorithm is generated randomly from the existing points in the research space [12] and [28]. After initialization, the fitness values of the population are evaluated, and the best candidate –which has the best fitness values –is introduced as a black hole and the other stars are selected as normal stars. Then, all of the stars commence moving towards the black hole and the black hole absorbs the stars around it. The movement of stars towards black hole is as follows:

$$x_i(t+1) = x_i(t) + rand * \left(x_{BH} - x_i\,(t)\right) \tag{1}$$

$$i = 1,2,3, \dots \dots \dots .. N$$

where $x_i(t+1)$ and $x_i\,(t)$ are the locations of the ith 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) [1].

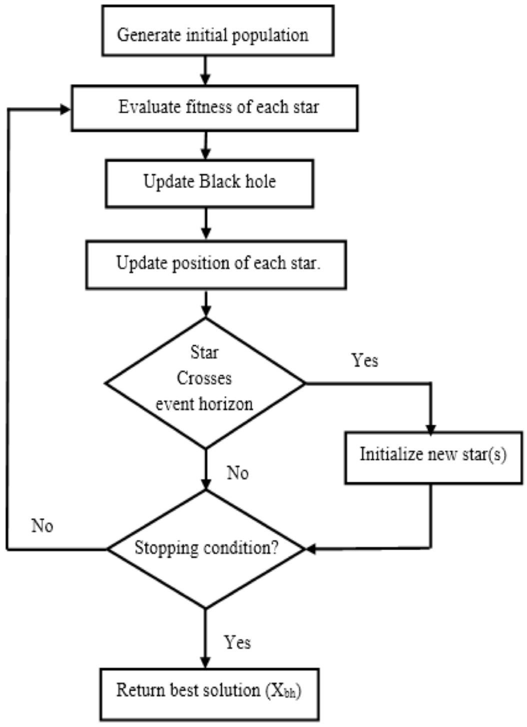

Figure 1: Flowchart of the standard BHA.

After the movement of the star to a new position (towards the black hole), if its fitness value is better than the value of the black hole, the star is selected as the black hole. Then, this algorithm continues with the black hole in the new location, and the stars start moving towards this new black hole. Moreover, there is a probability of crossing the event horizon (star's distance with black hole) during star movement towards the black hole [13] and [14].

A candidate solution (star) that crosses the event horizon of the black hole is swallowed by the black hole. Then, a new star following the swallowed one is generated and distributed randomly in the search space. This generation is performed to keep the number of stars (candidate solutions) constant. When all of the stars have moved once, the next iteration begins, as shown in Fig. 1. The radius of the event horizon (R) is formulated as follows:

$$\frac{f_{BH}}{\sum_{i=1}^{N} f_i} \tag{2}$$

wheref_{BH} is the fitness value of the black hole, *N* is the number of candidate solutions (stars) and fi is the fitness value of the ith star. When the star's distance to the black hole is less than a defined radius (R), the star is swallowed by the black hole.

3. GRAVITATIONAL SEARCH ALGORITHM

GSA is a heuristic optimization algorithm that has been gaining interest among the scientific community. GSA is a nature-inspired algorithm that is based on Newton's law of gravity and the law of motion [7]. GSA is grouped under the population-based approach and is reported to be more intuitive [9]. GSA was introduced by Rashedi et al. in 2009 and is intended to solve optimization problems. The population-based heuristic algorithm is based on the law of gravity and mass interactions [2].

The agents are considered as objects and their performance is measured by their masses. Gravity causes global movement, where all objects move towards objects with heavier masses. The slow movement of heavier masses guarantees the exploitation step of the algorithm and corresponds to good solutions. The law of gravity is shown in equation (3), and the law of motion in Equation (4).

$$F = G(M1M2)/R2 \quad (3)$$

$$a = {}^{F}/_{M} \quad (4)$$

F represents the magnitude of the gravitational force, *G* is the gravitational constant, *M1* and *M2* are the mass of the first and second objects and *R* is the distance between the two objects. Equation (3) shows that in Newton's law of gravity, the gravitational force between two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between the objects. For Equation (4), Newton's second law shows that when a force, *F*, is applied to an object, its acceleration, *a,* depends on the force and its mass, *M*.

In GSA, the agent has four parameters, which are the position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass represents the solution of the problem, where the gravitational and inertial masses are determined using a fitness function, as shown in fig. 2. The algorithm is navigated by adjusting the gravitational and inertia masses, whereas each mass presents a solution. Masses are attracted by the heaviest mass [10]. Hence, the heaviest mass represents an optimum solution in the search space. The steps of GSA are as follows:

Step 1: Agents initialization

The positions of the *N* number of agents are initialized randomly.

$$\left(x_i = \left(x_i^1 \ldots \ldots . x_i^d\right)\right) \qquad for\ i = 1,2, \ldots \ldots \ldots \ldots \ldots N \quad (5)$$

x_i^d represents the positions of the ith agent in the dth dimension, whereas *N* is the space dimension.

Step 2: Fitness evolution and best fitness computation

For minimization or maximization problems, the fitness evolution is performed by evaluating the best and worst fitness for all agents at each iteration.

Minimization problems:

$$best(t) = min\,fit\,j(t) \quad (6)$$

$$worst(t) = max\,fit\,j(t) \quad (7)$$

Maximization problems:

$$best\ (t) = max\,fit\,j(t) \quad (8)$$

$$worst(t) = min\,fit\,j(t) \quad (9)$$

fit j (t) represents the fitness value of the jth agent at iteration *t*; best (t) and worst(t) represent the best and worst fitness at iteration *t*.

Step 3: Gravitational constant (G) computation

Gravitational constant G is computed at iteration t.

$$G(t) = G_0\, e^{\,(-\,\varepsilon\alpha/T)} \quad (10)$$

G0 and *α* are initialized at the beginning and are reduced with time to control the search accuracy. T is the total number of iterations [2].

Step 4: Masses of the agents' calculation

The gravitational and inertia masses for each agent are calculated at iteration *t*.

$$M_{ai} = M_{pi} = M_{ii} = M_i, \; i = 1, 2, \dots \dots N \tag{11}$$

Mai and Mpi are the active and passive gravitational masses, respectively, and Mii is the inertia mass of the ith agent.

Step 5: Accelerations of agents' calculation

Acceleration of the i^{th} agents at iteration t is computed.

$$a_i^d(t) = F_i^d(t)/M_{ii}(t) \tag{12}$$

$F_i^d(t)$ is the total force acting on i^{th} agent calculated as:

$$F_i^d(t) = \sum rand_j \, F_{ij}^d(t) \tag{13}$$

$$j \in K_{best}, \; j \neq i$$

K_{best} is the set of first *K* agents with the best fitness value and biggest mass.

K_{best} will decrease linearly with time and at the end there will be only one agent applying force to the others.

$F_{ij}^d(t)$ is computed as the following equation:

$$F_{ij}^d(t) = G(t).\left(\frac{M_{pi}(t) * M_{ai}(t)}{R_{ij}(t)}(t) + \varepsilon\right).\left(X_j^d(t) - X_i^d(t)\right) \tag{14}$$

$F_{ij}^d(t)$ is the force acting on agent i from agent *j* at d^{th} dimension and t^{th} iteration.

$R_{ij}(t)$ is the Euclidian distance between two agents *i* and *j* at iteration *t. G(t)* is the computed gravitational constant at the same iteration while ε is a small constant.

Step 6: Velocity and positions of agents

Velocity and the position of the agents at next iteration *(t+1)* are computed based on the following equations:

$$V_i^d(t+1) = rand_i * V_i^d(t) + a_i^d(t) \tag{15}$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \tag{16}$$

Step 7: Repeat steps 2 to 6

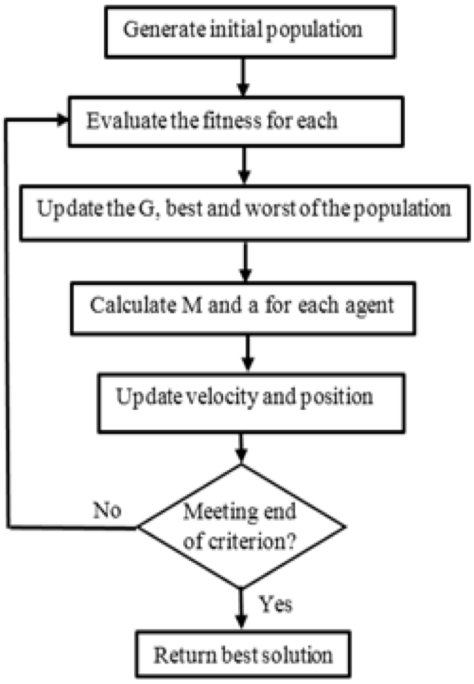

Figure 2: Steps of the standard GSA.

4. THE PROPOSED ALGORITHM (THE HYBRID ALGORITHM) (BH-GSA)

In GSA, there is a possibility of premature convergence. Because the guiding force attracts the objects to each other, the algorithm performance decreases if the objects converge to a non-optimal solution, and the algorithm suffers from slow searching speed in the last iterations.

In BHA, over time, exploring is reduced and the exploitation ability fades in, so the algorithm adjusts itself in the semi-optimal points. There should be a balance between exploration and exploitation to keep black hole algorithm safe from being trapped in local optima [12].

The Black Hole Algorithm can be improved by merging the two algorithms. The gravitational force between stars and the movement of stars to the black hole is adjusted while searching the solution space. To improve the standard Black Hole Algorithm (BHA) using (GSA), we assume some of the heavy objects are stars in a gravitational system, which become black holes, and the exploitation of GSA is improved.

In the duration of the algorithm, the radius of the black hole decreases, and more objects are encompassed, which helps to prevent premature convergence. Then, some of the best objects become the black hole, affecting other objects by their strong gravity. The other objects are divided into two categories: heavy agents and light agents. The position and velocity of the heavy agents and light agents change with (15) and (16), respectively. These influences depend on the distance of the black hole r to the other masses. This distance is compared with Schwarzschild radius Rs to determine whether the object is close enough to the black hole, and the principle of the hybrid algorithm is shown in Fig. 3.

5. EXPERIMENTS AND RESULTS

The hybrid BHGSA algorithm is applied to the 30 CEC 2014 test function to evaluate the performance of the algorithm, and by using MATLAB R2011, the test functions are analysed as minimization problems with the same search space for all functions, which is [-100, 100] for all dimensions [7].

The mean fitness of the algorithms for each test function was used to compare the results of all algorithms statistically. All the agents have fitness values which are evaluated by the fitness function, the mean fitness is the average over all members in the current population, which calculated after 2000 iterations and 50 runs.

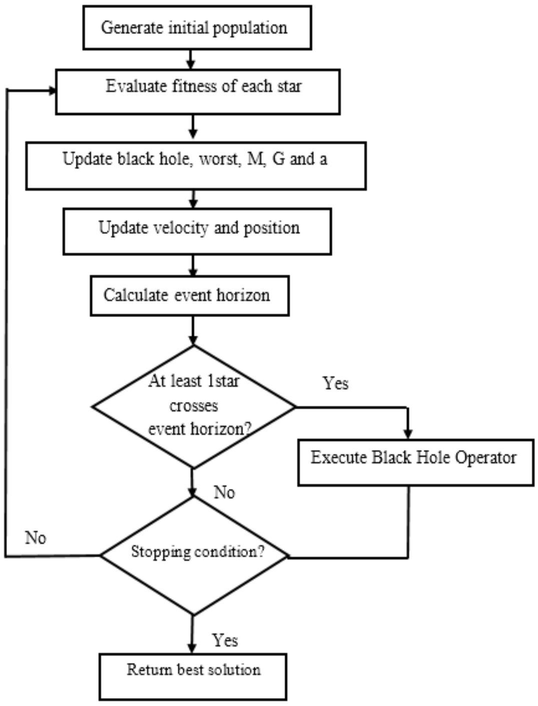

Figure 3: GSA-BH algorithm.

Table 1: Mean values of GSA-BH. BH and GSA.

Function	BHA	GSA	BH-GSA
F1	2472895000.25	340747493.45	**6186372.90**
F2	167571789177.31	388934724.62	**6653713.45**
F3	438046.16	156440.09	**122284.34**
F4	47843.71	4307.85	**2172.21**
F5	521.19	**519.99**	520
F6	673.85	661.90	**649.42**
F7	2332.59	1204.21	**727.21**
F8	1537.52	1071.13	**993.02**
F9	1795.57	1259.38	**1185.55**
F10	14790.94	9229.24	**7882.96**
F11	15029.09	10381.52	**8251.73**
F12	1203.84	**1200.07**	1200.24
F13	1308.89	1306.20	**1303.27**
F14	1834.30	1631.45	**1459.63**
F15	16257815.36	735631.42	**8483.01**
F16	1622.97	1622.64	**1621.75**
F17	204008925.69	398186063.45	**11078675.86**
F18	8869606986.12	7768759443.79	**11081025.87**
F19	3115.04	2757.25	**2044.235**
F20	343988.15	161537.83	**37005.71**
F21	46325593.56	25988841.19	**2711120.26**
F22	7248.82	47345.37	**4187.31**
F23	4040.82	**2518.30**	2838.22
F24	3089.08	**2620.69**	2622.95
F25	2924.45	**2707.05**	2728.17
F26	2724.74	2800.15	**2715.37**
F27	5169.06	4608.91	**4498.06**
F28	14138.96	**6245.14**	10448.59
F29	1061443110.45	**3100.21**	15322360.49
F30	12338450.86	4946907.33	**1520918.62**

All of the algorithms were subjected to the same parameter settings: the number of agents (N) was 100, the number of iterations (tMax) was 2000, the number of dimensions (D) was 50 and the number of runs (runMax) was 50. The bench-mark functions of CEC 2014 are summarized in Table 1.

The mean fitness of the proposed GSA-BH algorithm, GSA and BHA for each test function was used to compare the results of all algorithms statistically, as shown in table 1. These results are compared statistically to those obtained by the standard BH and GSA algorithms.

For unimodal test functions, it can be seen that the GSA-BH is better than GSA and BH for F1, F2 and F3 functions. For simple multimodal test functions, the results show that GSA-BH is better than GSA and BH for all cases, except in case of F12. For hybrid functions, the results of GSA-BH algorithm are better than GSA and BH in all cases. For the composition cases GSA-BH outperformed GSA and BH in F26 and F27 and was not able to outperform GSA and BH another cases.

6. CONCLUSIONS

In this paper, a hybrid BH-GSA is presented to solve different optimization problems with high accuracy. For this purpose, the Black Hole Algorithm and Gravitational Search Algorithm are combined. The combination of these algorithms has led to finding the global optimum with high accuracy. The results of the experiments on various test functions show that the new hybrid BH-GSA has higher performance in finding the centre of the search space by setting a local minimum trap at that position. Therefore, this algorithm can be used for optimization of different and complex projects.

ACKNOWLEDGEMENT

This research is funded by the Fundamental Research Grant Scheme (FRGS/1/2015/ICT02/MMU/03/1), which is awarded by Ministry of Higher Education Malaysia to Multimedia University.

REFERENCES

[1] Korany R. Mahmoud and Safwat Hamad. Parallel Implementation of Hybrid GSA-NM Algorithm for Adaptive Beam-Forming Applications. Progress in Electromagnetics Research B, 2014, Vol. 58, 47{57}.

[2] Harish Kundra and Harsh Sadawarti. Assortment of Nature Inspired Computing. International Journal of Advances in Science and Technology (IJAST).

[3] Deepthi S. and Aswathy Ravikumar. A Study from the Perspective of Nature-Inspired Metaheuristic Optimization Algorithms. International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 9, March 2015.

[4] Hatamlou, A., Black hole: A new heuristic optimization approach for data clustering. Inf. Sci. (NY), 222(2013), 175-184.

[5] Miroslaw Malek, Mohan GUruswamy, Howard Owens and Minhir Pandya, A Hybrid Algorithm Technique. Department of Computer Sciences, 1989, University of Texas at Austin.

[6] Shanhe Jiang, Yan Wanga and Zhicheng Jia. Convergence analysis and performance of an improved gravitational search algorithm. Applied Soft Computing 24 (2014) 363–384. Journal homepage: www.elsevier.com/locate/asoc.

[7] Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S., GSA: A Gravitational Search Algorithm. Inf. Sci. (NY), 179(2009), 2232-2248.

[8] Russel C. Elberhart and Yuhui Shi. Particle Swarm Optimization: Developments, Applications and Resources. 2001, IEEE.

[9] Mohamad Fadzli Haniffa, Hazlina Selamata and Salinda Buyamin, an Improved Global Particle Swarm Optimization for Faster Optimization Process, 2014, Jurnal Technology.

[10] Fister, I., Yang, X. S., Brest, J., and Fister D. A brief review of nature-inspired algorithms for optimization, 2013.

[11] Norlina Mohd Sabri, Mazidah Puteh and Mohamad Rusop Mahmood. An Overview of Gravitational Search Algorithm Utilization in Optimization Problems. IEEE 3rd International Conference on System Engineering and Technology, 19 - 20 Aug.

[12] Masoum Farahmandian and Abdolreza Hatamlou. Solving optimization problems using black hole algorithm. Journal of Advanced Computer Science & Technology, 4 (1) (2015) 68-74.

[13] Yuhui Shi. Particle Swarm Optimization Electronic Data Systems. Inc. Kokomo, in 46902, USA.

[14] Doddy Prayogo and Min-Yuan Cheng. Black Hole-Inspired Algorithm: An intelligent approach for solving constrained structural design optimization.

[15] Pier Luigi Bellon, Francesca Cantele, Sacha De Carlo and Salvatore Lanzavecchia. A trajectory-based algorithm to determine and refine Euler angles of projections in three-dimensional microscopy. Improvements and tests. ELSEVIER, March 2002.

[16] H. Omranpour, M. Ebadzadeh, S. Shiry and S. Barzegar. Dynamic Particle Swarm Optimization for Multimodal Function. International Journal of Artificial Intelligence (IJ-AI) Vol. 1, No. 1, March 2012, pp. ISSN: 2252-8938.

[17] J. P. Papa, A. Pagnin and S. A. Schellini. Feature Selection Through Gravitational Search Algorithm, 2011 IEEE 978-1-4577-0539.

[18] Badaruddin Muhammad, Zuwairie Ibrahim, Kamarul Hawari Ghazali, Kamil Zakwan Mohd Azmi, Nor Azlina Ab Aziz, Nor Hidayati Abd Aziz, and Mohd Saberi Mohamad, A New Hybrid Simulated Kalman Filter and Particle Swarm Optimization for Continuous Numerical Optimization Problems, ARPN Journal of Engineering and Applied Sciences, 2015, Vol. 10, No. 22, pp. 17171-17176.

[19] Thomas Osugi, Deng Kun and Stephen Scott. Balancing Exploration and Exploitation: A New Algorithm for Active Machine Learning. CSE Conference and Workshop Papers. Fifth IEEE International Conference on Data Mining, 2005.

[20] Mohamad Nizam Aliman, Zuwairie Ibrahim, Fardila Naim, Sophan Wahyudi Nawawi and Shahdan Sudin. Performance evaluation of Black Hole Algorithm, Gravitational Search Algorithm and Particle Swarm Optimization. Malaysian Journal of Fundamental and Applied Sciences, Vol.11, No.1 (2015), pp. 10-20.

[21] Seyed Ali Mirjalili, Siti Zaiton Mohd Hashim and Hossein Moradian Sardroud. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. ELSEVIER. Applied Mathematics and Computation 218 (2012) 11125–11137

[22] Seyedali Mirjalili and Siti Zaiton Mohd Hashim. 2010. A New Hybrid PSOGSA Algorithm for Function Optimization. International Conference on Computer and Information Application (ICCIA), pp.374-377.

[23] Natallia Kokash. An introduction to heuristic algorithms. Department of Informatics and Telecommunications, University of Trento, Italy, email: kokash@dit.unitn.it.

[24] Sahazati Md Rozali, Mohd Fua'ad Rahmat and Abdul Rashid Husain. Performance Comparison of Particle Swarm Optimization and Gravitational Search Algorithm to the Designed of Controller for Nonlinear System. Hindawi Publishing Corporation Journal of Applied Mathematics Volume 2014, Article ID 679435, 9 pages. http://dx.doi.org/10.1155/2014/679435

[25] Mohammad Doraghinejad, Hossein Nezamabadi-pour, Armindokht Hashempour Sadeghian and Malihe Maghfoori. A Hybrid Algorithm Based on Gravitational Search Algorithm for Unimodal Optimization.

[26] Mohammad Eskandarzadehalamdary, Behrooz Masoumi and Omid Sojodishijani. A New Hybrid Algorithm Based on black hole Optimization and Bisecting K-means for Cluster Analysis. The 22nd Iranian Conference on Electrical Engineering (ICEE 2014), May 20-22, 2014, Shahid Beheshti University.

[27] Cholavendhan Selvaraj, Siva Kumar R, Karnan M. A Survey on Application of Bio-Inspired Algorithms. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1), 366-370, 2014.

[28] Junqi Zhang, Kun Liu, Ying Tan, and Xingui He. 2008. Random black hole particle swarm optimization and its application. IEEE Int. Conference Neural Networks & Signal Processing Zhenjiang, China, June 8~10.

[29] Jian Jiaol, Xianjia Wang and Liubo Zhang. Applying Particle Swarm Optimization in Multiobjective Optimization and Hybrid Optimization. 2010 Second International Conference on Computational Intelligence and Natural Computing (CINC).

[30] Xin-She Yang, Shi Cheng. Hybrid Metaheuristic Algorithms Past, Present, and Future, Research Gate, January, 2015.

[31] A. A. Heidari and R. A. Abbaspour. A Gravitational Black Hole Algorithm for Autonomous UCAV Mission Planning in 3D Realistic Environments. International Journal of Computer Applications, Volume 95– No. 9, June 2014.

[32] Nisha, Simranjit Kaur and Sandeep Kumar Arora. Analysis of Black Hole Effect and Prevention through IDS in MANET. American Journal of Engineering Research (AJER) e-ISSN: 2320-0847 p-ISSN: 2320-0936 Volume-02, Issue-10, pp-214-220, 2013.

[33] Binjie Gu. and Feng Pan. Modified Gravitational Search algorithm with Particle Memory Ability and its application ICIC International. ISSN 1349-4198, Volume 9, Number 11, November, 2013.