mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 8 – 14
DOI: https://doi.org/10.15282/mekatronika.v1i2.4891

ORIGINAL ARTICLE

Improving Black Hole Algorithm using Gravitational Search, White Hole Operator, and Local Search

Suad Khairi Mohammed[1], Ahmad Afif Mohd Faudzi[2], Hamdan Daniyal[2], Badaruddin Muhammad[2], Nor Azlina Ab. Aziz[3], Zulkifli Md Yusof[4], Kamil Zakwan Mohd Azmi[4] and Asrul Adam[4]

[1]Faculty of Engineering, University of Technology, Baghdad, Iraq.
[2]Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.
[3]Faculty of Engineering and Technology, Multimedia University, 75450 Melaka, Malaysia.
[4]Faculty of Mechanical and Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.

ABSTRACT – Previously, the black hole (BH) algorithm has been subjected to various fundamental enhancements. Among others, white hole operator and local search have been embedded in the BH algorithm to improve its performance significantly. This paper shows that combination of gravitational search, white hole operator, and local search also able to improve the performance of the BH algorithm significantly.

ARTICLE HISTORY
Received: 13 January 2019
Accepted: 26 April 2019

KEYWORDS
Optimization
Black Hole
White Hole

Introduction

The black hole (BH) algorithm was introduced in 2013 [1]. It was originally intro-duced to solve clustering problems. The BH algorithm is called a population-based algorithm because many agents are used to find the solution to an optimization problem. The idea of the black hole is shown in Fig. 1. An event horizon is also shown. The event horizon is the boundary that covers the points of intense gravitational pull. If an object appears near the black hole (within the event horizon), the object is pulled towards the black hole due to the massive gravity of the black hole. In BH algorithm, if an agent is located within the event horizon, the agent will be pulled towards the black hole agent and disappears. Then, a new agent is re-initialized as shown in Fig. 2.

Interestingly, it was found that the BH algorithm is in fact a simplified version of PSO with inertia weights [2]. Technically, it was also proved that the BH algorithm is inferior to the performance of PSO with inertia weights. Nevertheless, the performance of the BH algorithm could be further improved. In 2016, Yaghoobi S. and Mojallali H. has proposed modified Black Hole algorithm by introducing genetic algorithm operators in order to improve optimization results[3]. Soto R. in 2018, to improve original BH has introduce Adaptive BH algorithm that able to dynamically adapts its population according to solving performance [4]. This is the reason why the BH algorithm is studied in this paper.

The Black Hole Algorithm

The BH algorithm is shown in Fig. 3. Since BH algorithm is a population-based algorithm, N number of agents are needed. Let d denotes the number of dimensions for an optimization problem, a solution, X, in a search space is kept by an agent i at iteration t as follows:

$$X_i(t) = \left(X_i^1(t), X_i^2(t), \dots, X_i^d(t)\right) \quad (1)$$

The BH algorithm begins with initialization where a randomly generated population of candidate solutions are placed in the search space. For each agent i, the initial solution can be represented as:

$$X_i(0) = \left(X_i^1(0), X_i^2(0), \dots, X_i^d(0)\right) \quad (2)$$

After the initialization, the fitness values of the population are evaluated. The best agent, which has the best fitness value, is chosen as the black hole while other agents are selected as normal agents. For the case of function minimization problems, during initialization, the black hole agent is determined as follows:

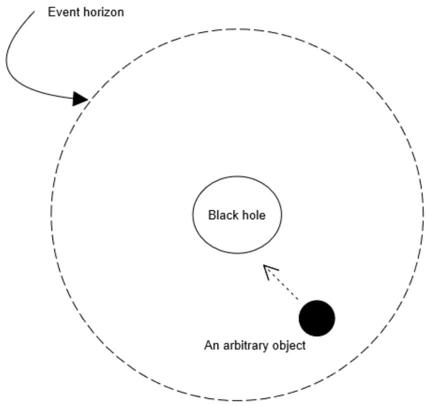

Figure 1. An illustration of a black hole event horizon of a black hole.

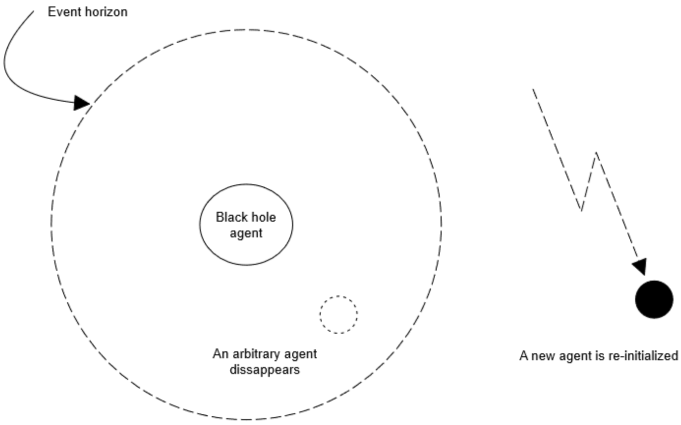

Figure 2. The agent near the black hole disappears and a new agent is re-initialized.

$$BH = \min_{i\in\{1,\dots,N\}} fit_i(t)|_{t=0} \quad (3)$$

The black hole agent keeps the best-so-far solution, XBH. The best-so-far solution is different than the best solution. The best solution is defined as the best solution obtained at specific iteration, t. On the other hand, the best-so-far solution is the best solution found from the initial iteration, $t = 0$, until current iteration, t. Hence, for $t \neq 0$, an agent i is selected as the black hole agent if the fitness value of that agent, f_i, is better than the fitness value of the black hole agent, f_{BH}. Specifically, for the case of function minimization, $f_i < f_{BH}$.

Once the black hole agent and normal agents are identified, the radius of the event horizon, RBH, is formulated as follows:

$$R_{BH} = \frac{f_{BH}}{\sum_{i=1}^{N} f_i} \quad (5)$$

where f_{BH} is the fitness value of the black hole agent, N is the number of agents, and fi is the fitness value of the i^{th} star.

The next step is solution update, which is applied to all agents except the black hole agent. Other than black hole agent, the agents can be categorized into two groups. The first group of agents is the agents located within the event horizon. This agent will be swallowed by the black hole agent. Then, a new agent following the swallowed one is generated and distributed randomly in the search space. This generation is to keep the number of agent constant. The second group of agents are agents located far from the black hole agent. In other words, these agents are not within the event horizon. These agents move towards the black hole agent and the updated solution can be computed as follows:

*CORRESPONDING AUTHOR | Suad Khairi Mohammed | ✉ suad.khairim@gmail.com

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 8 – 14
DOI: https://doi.org/10.15282/mekatronika.v1i2.2377

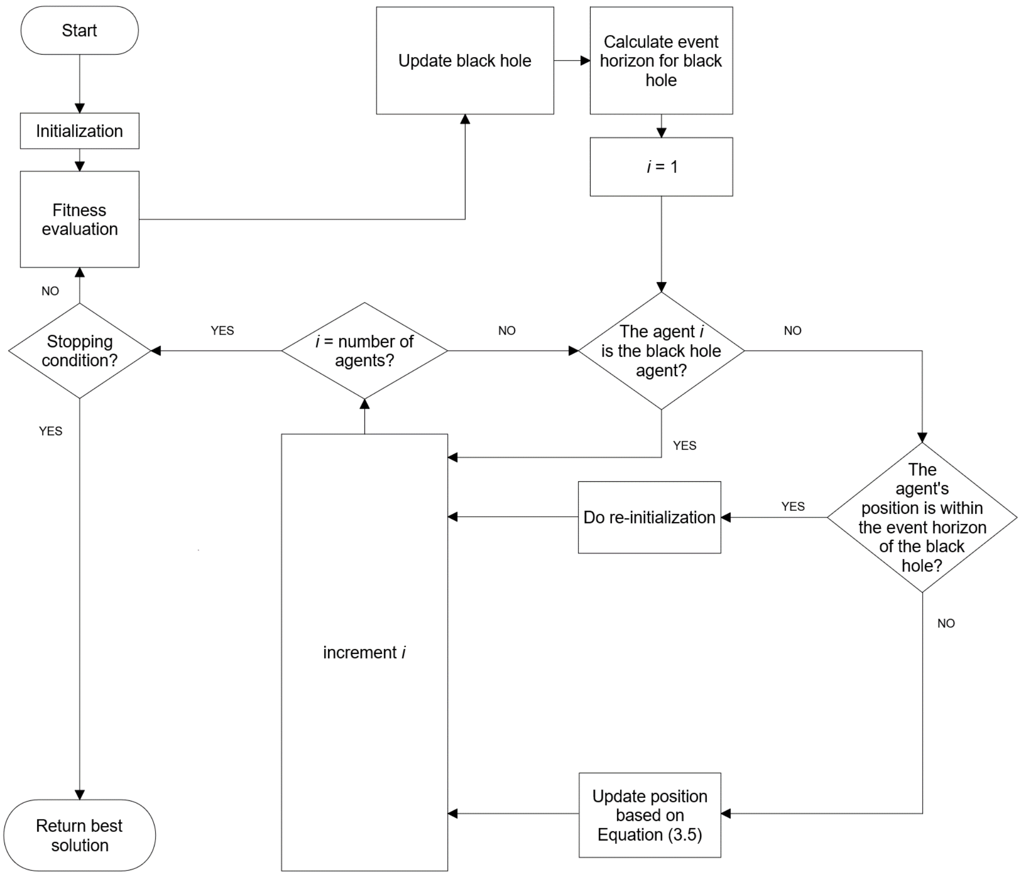

Figure 3. Flowchart of the black hole algorithm.

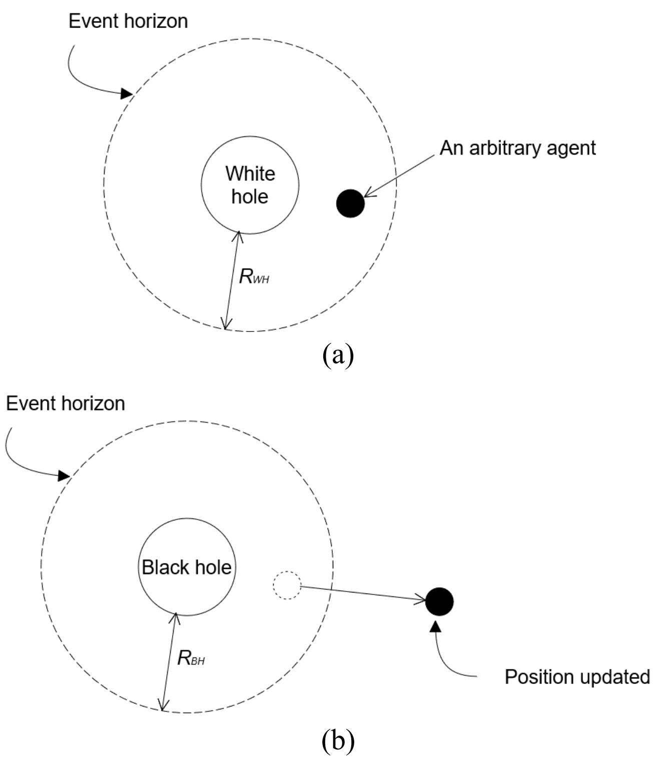

Figure 4. An illustration of the concept of white hole operator.

*CORRESPONDING AUTHOR | Suad Khairi Mohammed | ✉ suad.khairim@gmail.com

$$X_i(t+1) = X_i(t) + rand \times (X_{BH} - X_i(t)) \quad (5)$$

where $X_i(t+1)$ and $X_i(t)$ are the locations of the i^{th} agent at iterations $t+1$ and t, respectively. The *rand* is a random number belonging to [0,1] and X_{BH} is the location of the black hole agent. After all the agents have updated their position, the next iteration begins if the termination condition is not met. Otherwise, the best-so-far, X_{BH}, solution is reported.

Improving the Black Hole Algorithm

Among others, improvements of the BH algorithm using gravitation search [5], local search [6], and white hole operator [7] have been reported in literature.

White Hole Operator

As oppose to black hole agent in the BH algorithm, the white hole can be assigned to the worst agent in the population. Hence, the white hole is updated as follows:

$$WH = \max_{i \in \{1,\ldots,N\}} fit_i(t) \quad (6)$$

Also, the white hole has its own event horizon and the radius of the event horizon, R_{WH}, which can be calculated based on the following equation:

$$R_{WH} = \frac{fit_{WH}}{\sum_{i=1}^{N} fit_i} \quad (7)$$

where fit_{WH} is the fitness value of the white hole, N is the number of agents, and fit_i is the fitness value of the i^{th} star.

As shown in Fig. 4(a), an arbitrary agent i could be updated to a position in the search space within the event horizon of the white hole. In this case, the agent is pushed by the white hole, as shown in Fig. 4(b). Due to this, the position of the agent i is updated as follows:

$$X_i(t+1) = X_i(t) + rand \times (X_{WH} + X_i(t)) \quad (8)$$

where $X_i(t+1)$ and $X_i(t)$ are the locations of the arbitrary agent i at iterations $t+1$ and t, respectively. The *rand* is a random number belonging to [0,1] and X_{WH} is the location of the white hole agent.

The Gravitation Search

Gravitation is a natural phenomenon in which all objects with mass are brought to-ward one another. This principle has been used in gravitation search algorithm [8]. Consider two objects which are separated by a distance R. The amount of gravity that something possesses is proportional to its mass and distance between it and an-other object. As distance from the object increases, the gravitational force reduces which means less gravitational attraction. The most basic formulation of force is shown in (9), which represents the gravitational force between two objects, M_i and M_j.

$$F_{ij} = G\frac{M_j \times M_i}{R^2} \quad (9)$$

where G is the gravitational constant, R is the distance separating two objects, and M_i and M_j represent the mass of object i and object j, respectively.

Local Search

The basic idea of the local search is to find neighbourhood solution around the best solution. Following the previous study [6], not all the agents are subjected to local search. The white hole agent that keep the worst solution at the iteration, t, is selected and the local search is applied to the white hole agent. Let $X_i^d = X_{worst}^d = X_{WH}^d$, the local search is applied to every dimension, d, based on (10).

$$X_i^d(t+1) = X_{BH}^d(t) + rand_d \times e^{-5t/T} \quad (10)$$

Where $X_i^d(t+1)$ is the solution after the local search is applied, X_{BH} is the location of the black hole agent, t is the iteration number, T is the maximum number of iterations, and $rand_d \in [0,1]$ is a random number, which is generated at every dimension.

The Gravitational Black Hole White Hole Local Search (GBHWHLS) Algorithm

The GBHWHLS algorithm, as shown in Fig. 5, combines all the elements of black hole, gravitational search, white hole operator, and local search. The black hole agent is not subjected to any position update. Agent that is located within the event horizon of the black hole is re-initialized. The worst agent is subjected to the local search based on (10). Agent that is located within the event horizon of the white hole is sub-jected to position update based on (8). If an agent is not a black hole or white hole agent, at the same time it is not the worst agent, and it is not located within the event horizon of the black hole and white hole, that agent updates its position based gravitational search algorithm [1] as follows:

*CORRESPONDING AUTHOR | Suad Khairi Mohammed | ✉ suad.khairim@gmail.com

VOL. 01, ISSUE 02, 8 – 14
DOI: https://doi.org/10.15282/mekatronika.v1i2.2377

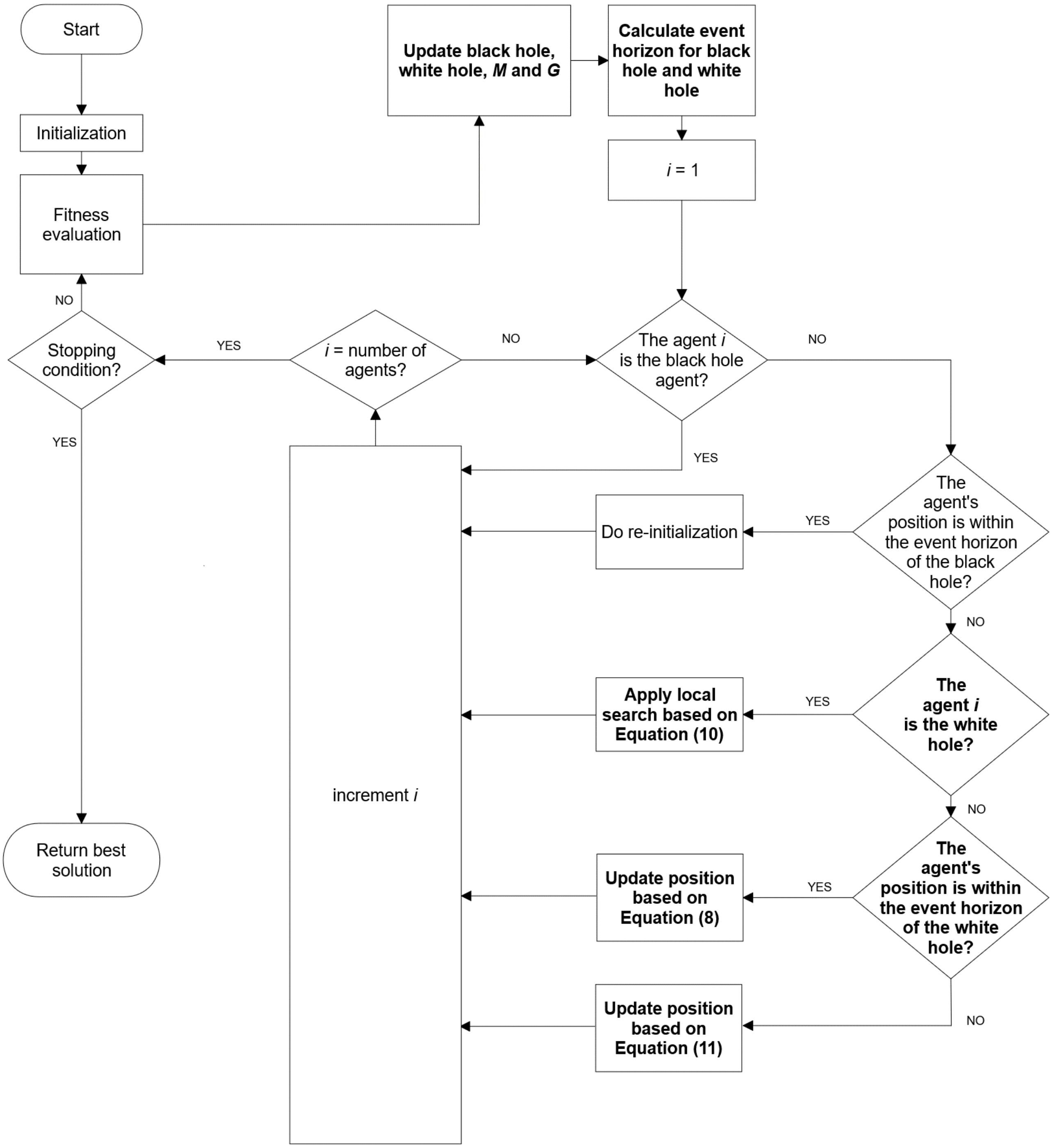

Figure 5. The flowchart of the GBHWHLS algorithm.

$$X_i^d(t+1) = X_i^d(t) + rand_i \times V_i^d(t) + a_i^d(t) \quad (11)$$

where the acceleration, $a_i^d(t)$, is calculation using the following equation:

$$a_i^d(t) = \frac{\sum rand_j F_{ij}^d(t)}{M_i(t)} \quad (12)$$

$$F_{ij}^d(t) = G_0 e^{-20t/T} \frac{M_i(t) \times M_j(t)}{R_{ij}(t)+\varepsilon} \times \left(X_j^d(t) - X_i^d(t)\right) \quad (13)$$

where G_0 is a constant and T is the total number of iterations. The agent in the population has its own mass which is calculated as follows:

*CORRESPONDING AUTHOR | Suad Khairi Mohammed | ✉ suad.khairim@gmail.com

Table 1. The experimental parameters.

Parameter	Value
Number of iterations (*T*)	10,000
Number of runs	51
Number of agents	100
Dimensions	50
Search space	[-100 100]
G_o	100

Table 2. The mean value of BH vs GBHWHLS (Function 1-3).

Function	BH	GBHWHLS	Optimal Fitness
1	5611014	**1595898**	100
2	4997329	**7518**	200
3	**14041**	23478	300

Table 3. The mean value of BH vs GBHWHLS (Function 4-16).

Function	BH	GBHWHLS	Optimal Fitness
4	609	**523**	400
5	520	**519**	500
6	658	**637**	600
7	701	**700**	700
8	**953**	1038	800
9	1249	**1201**	900
10	**3816**	5797	1000
11	8308	**7791**	1100
12	1200	**1200**	1200
13	1300.5	**1300.4**	1300
14	**1400.26**	1400.29	1400
15	1810	**1508**	1500
16	**1621**	1622	1600

Table 4. The mean value of BH vs GBHWHLS (Function 17-22).

Function	BH	GBHWHLS	Optimal Fitness
17	639170	**148858**	1700
18	**2476**	4081	1800
19	1960	**1940**	1900
20	**9023**	10954	2000
21	429192	**116151**	2100
22	3786	**3356**	2200

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad (14)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{BH(t) - worst(t)} \quad (15)$$

where *best*(*t*) and *worst*(*t*) represent the best and worst fitness at iteration *t*, respectively.

Table 5. The mean value of BH vs GBHWHLS (Function 22-30).

Function	BH	GBHWHLS	Optimal Fitness
23	2652	**2598**	2300
24	2665	**2610**	2400
25	2749	**2700**	2500
26	2796	**2794**	2600
27	4729	**4356**	2700
28	11732	**11430**	2800
29	**10839**	469094	2900
30	69850	**59603**	3000

Experiment, Result, and Statistical Analysis

In this research, the algorithms are tested using the CEC2014 benchmark suites, which consists of 3 unimodal functions, 13 simple multimodal functions, 6 hybrid functions, and 8 composite functions [9]. The experimental parameters are tabulated in Table 1.

The mean accuracy of BH and GBHWHLS algorithms are tabulated in Table 2, Table 3, Table 4, and Table 5. Minimum average fitness is written in bold and the optimal fitness is also shown.

For statistical analysis, Wilcoxon signed rank test [10] was used. The Wilcoxon test usually is used when the population cannot be assumed to be normally distributed or it can be used to compare two related samples, matched samples, or repeated meas-urements on a single sample to assess whether their population mean ranks differ. Particularly, the null hypothesis for the test assumes that there is no significant differ-ence between the mean error values of test algorithms while the alternative hypothe-sis tries to determine if there is a significant difference between test algorithms using 10% ($\alpha = 0.1$) significance level. Since the number of samples is 30, the critical value for the test is equal to 152. The sum of ranks where the BH algorithm outperforms the competing algorithm is denoted as R- while the sum of ranks where the BH algo-rithm is outperformed by the competing algorithm is denoted as R+. Hence, the com-peting algorithm is better than the BH algorithm if R+ > R- and the competing algo-rithm is significantly better than the BH algorithm if R- value is less than the critical value. Based on the mean accuracy of BH and GBHWHLS, The R+ and R- are 330 and 135, respectively. Hence, Wilcoxon test result shows that the GBHWHLS algorithm is significantly better than the BH algorithm.

Conclusions

The BH algorithm is inspired by the black hole phenomena. Since the introduction of the BH algorithm, a lot of studies have been conducted to improve the BH algorithm fundamentally. In this paper, fundamental improvement to the BH algorithm is performed based on the combination of gravitation search, white hole operator, and local search. Experimental result has shown that the GBHWHLS algorithm able to outperform the BH algorithm in 22 test functions out of 30 test functions of CEC2014 benchmark suite. Statistical analysis also shows that the GBHWHLS algorithm is significantly better than the BH algorithm. The next step of this research is to benchmark the GBHWHLS algorithm with other algorithms such as particle swarm optimization.

Acknowledgement

The authors are indebted to the Universiti Malaysia Pahang for providing a financial support (RDU1703224) to this research.

References

[1] Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. Information Sciences, 222, 175-184.

[2] Piotrowski, A.P., Napiorkowski, J.J. and Rowinski, P. M. (2014). How novel is the "novel" black hole optimization approach?. Information Sciences, 267, 191-200.

[3] Yaghoobi S. and Mojallali H. (2016). Modified black hole algorithm with genetic operators. International Journal of Computational Intelligence Systems., vol 9, issue 4, pp. 652 – 665.

[4] Soto, R., Crawford B., Olivares R., Taramasco C., Figueroa I., Gomez A., Castro C. and Paredes F. (2018). Adaptive black hole algorithm for solving the set covering problem. Mathematical Problems in Engineering, vol. 2018, pp. 1-23.

[5] Heidari, A. and Abbaspour, R. (2014). A gravitational black hole algorithm for autonomous UCAV mission planning in 3D realistic environments. International Journal of Computer Applications, vol. 95, no. 9.

[6] Mohammed, S.K., Ibrahim, Z., Daniyal, H., Subari, N., Muhammad, B., Md Yusof, Z., Ab Aziz, N.A. and Mohd Azmi, K.Z. (2017). Improving the effectiveness of the black hole algorithm using a local search technique. International Journal of Simulation: Systems, Science and Technology, vol. 18, no. 4, pp. 12.1-12.5.

[7] Muhammad, B., Mohammed, S.K., Pebrianti, D., Abdul Ghani, N.M., Abdul Aziz, N.H., Ab Aziz, N.A., Shapiai, M.I., Subari, N., Ab Rahman, T., Mohd Azmi, K.Z. and Ibrahim, Z. (2018). A black hole algorithm with white hole operator and its application in parameter tuning of proportional-integral-derivative controller. SICE International Symposium on Control Systems.

[8] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information Sciences, vol. 179, no. 13, pp. 2232-2248.

[9] Liang, J., Qu, B., Suganthan, P. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.

[10] Derrac, J., García, S., Molina, D., Herrera, F. (2011). A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 3-18.