Black hole white hole algorithm with local search

Zuwairie Ibrahim, Suad Khairi Mohammed, Norazian Subari, Asrul Adam, Zulkifli Md Yusof
Universiti Malaysia Pahang, Pahang, Malaysia

Nor Azlina Ab Aziz, Nor Hidayati Abdul Aziz
Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia

Tasiransurini Ab Rahman
Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

Norrima Mokhtar
Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia

Email: zuwairie@ump.edu.my, suad.khairim@gmail.com, aziansubari@ump.edu.my, asrul@ump.edu.my, zmdyusof@ump.edu.my, azlina.aziz@mmu.edu.my, hidayati.aziz@mmu.edu.my, surini@uthm.edu.my, norrimamokhtar@um.edu.my

Abstract

Black hole algorithm (BHA) is an optimization algorithm inspired by the black hole discovery in relativity theory. Recently, white hole operator, which is based on the opposite of black hole, has been introduced in BHA. In this paper, a local is added in the BHA with white hole operator.

Keywords— black hole algorithm, fundamental, optimization

1. Introduction

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 strong gravity.

The BHA algorithm (BHA) [1] is a population-based meta-heuristic algorithm inspired by the physical phenomenon of black hole. In BHA, the agent with the best solution mimics the black hole. The event horizon is calculated and any agent within the event horizon vanishes and re-initialized in the search space.

The BHA is shown in Figure 1. Let N is the number of agents and d is the number of dimension for an optimization problem, a solution, X, in a search space is kept by an agent i at iteration t as follows:

$$X_i(t) = \left(X_i^1(t), X_i^2(t), \ldots, X_i^d(t)\right) \tag{1}$$

The BHA 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:

$$X_i(0) = \left(X_i^1(0), X_i^2(0), \ldots, X_i^d(0)\right) \tag{2}$$

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:

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

In this study, the black hole agent keeps the best-so-far solution, X_{BH}. 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, $f_i < f_{\mathrm{BH}}$.

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

© The 2018 International Conference on Artificial Life and Robotics (ICAROB2018), Feb. 1-4, B-Con Plaza, Beppu, Oita, Japan

$$R_{BH} = \frac{f_{BH}}{\sum_{i=1}^{N} f_i} \tag{4}$$

where f_{BH} is the fitness value of the black hole agent, N is the number of agents, and f_i 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:

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

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. This solution update can be summarized in the Pseudocode 1.

PSEUDOCODE 1: Solution update in BHA

if agent i^{th} position is within the event horizon
then
do re-initialization
if agent i^{th} position is not within the event horizon
then
update the position based on Eq. (5)
else
end

After all the agents have updated their position, the next iteration begins if the termination criteria is not met. Otherwise, the best-so-far, X_{BH}, solution is reported.

2. The black hole white hole (BHWH) algorithm

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

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

Also, similar to the black hole, the white hole has its own event horizon and the radius of the event horizon, *RWH*, can be calculated based on the following equation:

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

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 *ith* star.

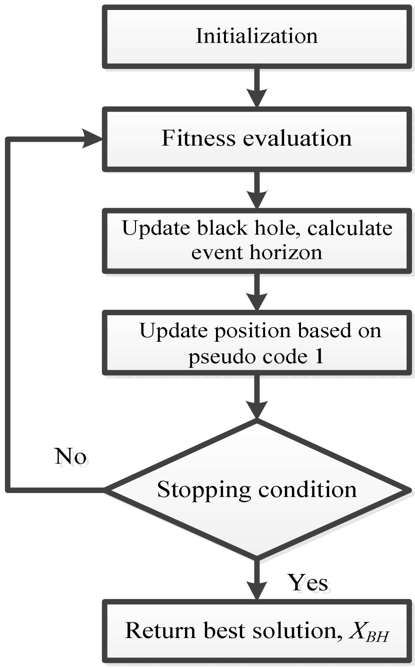

Fig. 1. Flowchart of BHA.

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. Due to this, the position of the agent i is updated as follows:

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

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.

Figure 2 shows the flowchart of the black hole algorithm with white hole operator, which is also termed as BHWH algorithm [2]. The difference between the proposed BHWH algorithm with the BHA shown in Figure 1 is the inclusion of a white hole agent as the worst solution. Note that there are two kind of event horizons. These are the event horizon for black hole and white hole agents and should be calculated at every iteration. Also, due to the inclusion of white hole agent which pushes any nearby agent, the position update in BHWH algorithm can be explained in Pseudocode 2.

3. The proposed BHWH algorithm with local search

The basic idea of the local search is to find neighbourhood solution around the best solution. In this study, 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.

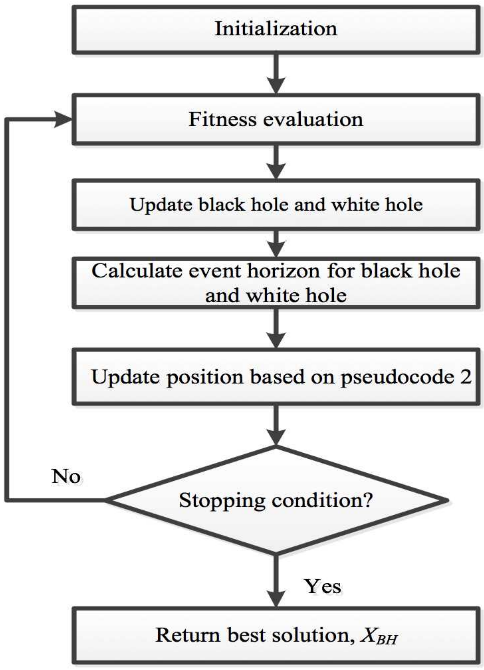

Fig. 2. Flowchart of BHWH algorithm.

Table 1. Experimental setting parameter

Iteration	10000
Runs	51
Agents	100
Dimension	50
Search Space	[-100 100]

PSEUDOCODE 2: Solution update in BHWH algorithm

if agent *ith* is black hole agent **then**
position not updated
if agent *ith* position is within the event horizon of black hole **then**
do re-initialization
if agent *ith* position is within the event horizon of white hole **then**
update the position based on Equation (8)
else
update the position based on Equation (5)
end

Let $X_i^d = X_{worst}^d = X_{WH}^d$, the local search is applied to every dimension, *d*, based on Equation (9).

$$X_i^d(t+1) = X_{BH}^d(t) + rand_d \times e^{-5t/T_{max}} \tag{9}$$

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_{max} is the maximum number of iteration, and $rand_d \in [0,1]$ is a random number, which is generated at every dimension.

The local search is applied to the worst agent in the population. It is possible to select any worst solution at iteration *t* during an implementation of an optimization algorithm and apply Equation (9) to the worst agent. In this study, the local search is incorporated in the BHWH algorithms. As a result, a variant of BH algorithm can be developed, which is called black hole white hole (BHWH) algorithm with local search (BHWHLS). The flowchart of BHWHLS algorithm is similar to the flowchart of BHWH algorithm in Figure 2. However, considering the local search, the solution for BHWHLS algorithm is updated according to Pseudocode 3, which replaces the Pseudocode 2.

PSEUDOCODE 3: Solution update in BHWHLS algorithm

if agent *ith* is black hole agent **then**
position not updated
if agent *ith* position is within the event horizon of black hole **then**
do re-initialization
if agent *ith* is the white hole **then**
apply local search based on Equation (9)
if agent *ith* position is within the event horizon of white hole **then**
update the position based on Equation (8)
else
update the position based on Equation (5)
end

4. Experiment, result, and discussion

The experiments in this study were implemented based on the parameter setting tabulated in Table 1. The performance of the BHWHLS algorithm and the original BHA were studied by solving CEC2014 benchmark functions [3]. The results are tabulated in Table 2. The values in bold indicate the smaller or better result. Table 2 shows that the BHWHLS algorithm outperforms BHA for all the unimodal functions, while for simple multimodal functions, the accuracy of BHWHLS algorithm is very good, except for Function 14. For hybrid functions, the BHWHLS algorithm reaches to optimum solution for all functions except for Function 18 and Function 19. Finally, even though the BHWHLS algorithm performs BHA in most of the composition functions, the accuracy are quite unsatisfactorily for the case of Function 24 and Function 25. An example of convergence curve is shown in Figure 3.

Acknowledgement

This research is funded by the Universiti Malaysia Pahang (RDU1703224).

References

[1] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering", *Information Sciences*, 222, 175-184, 2012.
[2] S.K. Mohammed, Z. Ibrahim, H. Daniyal, and N.A. Ab. Aziz, "White Hole-Black Hole Algorithm", *The National Conference for Postgraduate Research*, 824-833, 2016.

© The 2018 International Conference on Artificial Life and Robotics (ICAROB2018), Feb. 1-4, B-Con Plaza, Beppu, Oita, Japan

Table 2. Performance comparison between the BHA and BHWHLS

Type of Function	Function ID	The ideal value	BHA (Mean value)	BHWHLS (Mean value)
Unimodal	F1	100	5611014.427	**2395827.785**
	F2	200	4997329.195	**8365.062195**
	F3	300	14041.12327	**6624.351763**
Simple multimodal	F4	400	609.8508916	**549.1541895**
	F5	500	520.0161923	**520.0000065**
	F6	600	658.7739009	**657.1073197**
	F7	700	701.1662029	**700.0187492**
	F8	800	953.499765	**928.9684462**
	F9	900	1249.316564	**1221.594196**
	F10	1000	3816.305871	**3214.544468**
	F11	1100	8308.348714	**7890.732838**
	F12	1200	1200.797984	**1200.757164**
	F13	1300	1300.56279	**1300.492432**
	F14	1400	**1400.261316**	1400.308738
	F15	1500	1810.089933	**1722.79273**
	F16	1600	1621.682464	**1621.5487**
Hybrid	F17	1700	639170.063	**235277.876**
	F18	1800	**2476.577727**	3470.864961
	F19	1900	**1960.011302**	1965.653521
	F20	2000	9023.306146	**4135.276728**
	F21	2100	429192.2267	**187917.412**
	F22	2200	3786.065395	**3758.516987**
Composition	F23	2300	2652.810511	**2645.194641**
	F24	2400	**2665.506218**	2668.43428
	F25	2500	**2749.939581**	2753.661216
	F26	2600	2796.226359	**2702.390476**
	F27	2700	4729.276014	**4655.848694**
	F28	2800	11732.29457	**10898.23985**
	F29	2900	10839.35982	**8746.81227**
	F30	3000	69850.72279	**37593.04308**

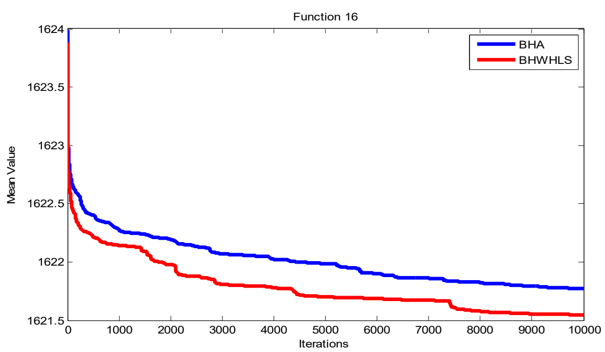

Fig. 3. Example of convergence curve (Function 16).

[3] J. J. Liang, B-Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization", *Technical Report 201311*, *Computational Intelligence Laboratory*, *Zhengzhou University*, Zhengzhou, China and *Technical Report*, *Nanyang Technological University*, Singapore, December 2013.