Improving the Effectiveness of the Black Hole Algorithm using a Local Search Technique

Suad Khairi Mohammed, Hamdan Daniyal, Norazian Subari, Badaruddin Muhammad, Zulkifli Musa
Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang
Pekan, Pahang, Malaysia
suad.khairim@gmail.com, zuwairie@ump.edu.my, hamdan@ump.edu.my, aziansubari@ump.edu.my, badaruddinmuhammad@yahoo.com, zkifli@ump.edu.my

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

Zuwairie Ibrahim, Kamil Zakwan Mohd Azmi
Faculty of Manufacturing Engineering
Universiti Malaysia Pahang
Pekan, Pahang, Malaysia
kamil_zakwan@yahoo.com.my

Tasiransurini Ab Rahman
Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia
Johor, Malaysia
surini@uthm.edu.my

***Abstract* - The main objective of the proposed algorithm in this paper is to modify the BH algorithm such that the simplicity can be resolved and improvements can be obtained in solving numerical optimization problems. The modified BH algorithm is presented by apply local search in BH algorithm to find neighborhood solution around the best solution. The black hole with local search (BHLS) algorithm is used as a new optimization approach to solve numerical optimization problems, specifically, unimodal, multimodal, hybrid, and composite functions of CEC2014 test suite.**

Keywords - Meta-heuristic, black hole, local search and numerical optimization problems.

I. INTRODUCTION

Almost all new meta- heuristics algorithms can be referred to the nature-inspired. The nature inspired algorithms lies in the fact that it receives its sole inspiration from nature. They have the ability to describe and resolve complex relationships from intrinsically very simple initial conditions and rules with little or no knowledge of the search space. Nature is the perfect example for optimization, because if we closely examine each and every features or phenomenon in nature it always find the optimal strategy.

Some of these meta-heuristic algorithms were inspired by nature such as ant colony optimization (ACO) [1], bee colony optimization [2], particle swarm optimization (PSO) [3], gravitational search algorithm (GSA) [4], and black hole algorithm (BHA) [5] with additional features that allow them to explore the entire search space. Meta-heuristics are typically high-level strategies which guide an underlying, more problem specific heuristic, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descents by allowing the local search to escape from local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more intelligent way than just providing random initial solutions. These algorithms are stochastic and approximate. They are stochastic in the sense that they try different random solutions through the search, and they are approximate since they try only a subset of the search space. Thus, the-meta-heuristic algorithms [6] cannot guarantee an optimal solution, but in most cases they result in a near optimal solution. Every meta-heuristic algorithm has mainly two components. First one is exploration and other one is exploitation. Exploration is the process of visiting entirely new regions of a search space, whilst exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points [7]. So the advantage of these meta-heuristic is that the search in a smart way, and they balance between exploitation (which is focusing the search in the neighborhood of the best solution) and exploration (which is searching in the entire research space). This smart balance will help to find a near optimal solution in less time and it will also avoid these meta-heuristic algorithms from getting trapped into in a local optimal solution.

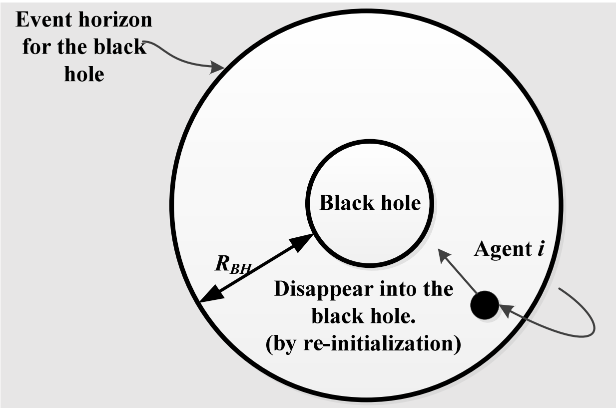

Figure 1. The event horizon of the black hole.

In solving of complex, multimodeal, high dimensional and nonlinear problems; the meta-heuristic optimization methods are used [8]. Generally, these problems can be seen in engineering, industry, business and many other areas. Scientists utilize several physical, chemical, biological laws which are helped them to improve new optimization methods. The organization of the paper is as follows: Section II provides the concept of black hole algorithm. The flowchart of black hole algorithm is discussed in Section III. Section IV introduces the local search method. Section V presents the proposed black hole with local search algorithm. The results and discussion are offered in section VI. The conclusion is drawn in Section VII.

II. CONCEPT OF BLACK HOLE ALGORITHM

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. The idea of the black hole is shown in Fig. 1. The R_{BH} determines the event horizon for the black hole. If a star appears near the black hole (within the event horizon), the star is pulled towards the black hole due to the massive gravity of the black hole. Black Hole (BHA) algorithm [5] is a population-based meta-heuristic algorithm inspired by the physical phenomenon of black hole. In BH algorithm, 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.

III. THE FLOWCHART OF BLACK HOLE ALGORITHM

The black hole (BH) algorithm is shown in Fig. 2. Since BH algorithm is a population-based algorithm, N number of agents are needed. Let d as 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:

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

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:

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

In this study, the black hole agent actually 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 is better than the fitness value of the black hole agent. Specifically, for the case of function minimization, $f_{x_i} < f_{BH}$.

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

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

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 *ith* 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 agent 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

DOI 10.5013/IJSSST.a.17.04.12

ISSN: 1473-804x online, 1473-8031 print

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:

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

where $X_i(t+1)$ and $X_i(t)$ are the locations of the ith 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. After all the agents have updated their position, the next iteration begins if the termination criteria is not met. Otherwise, the best-so-far, XBH, solution is reported.

PSEUDOCODE 1: Solution update in black hole algorithm

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

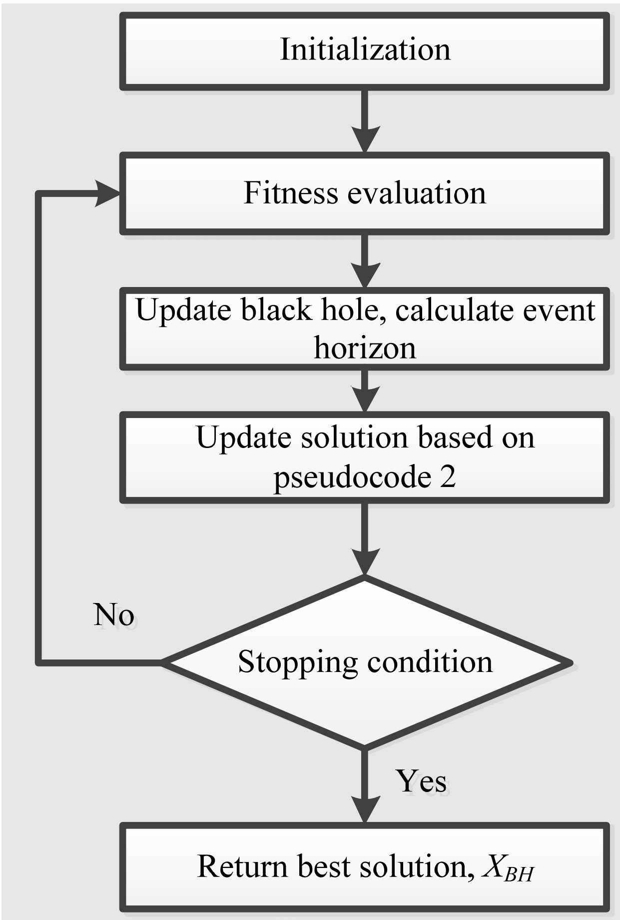

Figure 3. The black hole with local search (BHLS) algorithm.

IV. THE LOCAL SEARCH METHOD

The basic idea of the local search is to find neighbourhood solution [9] around the best solution [10-11]. The BH algorithm is a population-based optimization algorithm. However, not all the agents are subjected to local search. To apply local search in BH algorithm, the worst agent that keep the worst solution, X_{worst}, at the iteration, t, is selected and the local search is applied to the worst solution, X_{worst}. In this study, the local search is applied to every dimension, d, based on and the updated solution after the local search is applied is called:

$$X_{LS}^i(t) = X_{BH}^i(t) + rand_i \times e^{-5t/T_{max}} \quad (6)$$

where X_{LS} 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 $rand_i \in [0.1]$ is a random number, which is generated at every dimension.

V. THE PROPOSED BLACK HOLE WITH LOCAL SEARCH ALGORITHM

The black hole with local search (BHLS) algorithm is shown in Fig. 3. Most of the steps are similar to the BH algorithm. By considering the local search, which is applied to the worst solution, the solution update is modified as shown in pseudocode 2. For the remaining of the agents, solution update is applied based on either by re-initialization or according to Eq. (6), depending on the location of the agent.

PSEUDOCODE 2: Solution update in black hole with local search algorithm

if agent *i*th is the worst agent **then**
    do local search based on Eq. (6)
if agent *i*th's position is within the event horizon **then**
    do re-initialization
if agent *i*th's position is not within the event horizon **then**
    update the position based on Eq. (5)
else
end

VI. RESULTS AND DISCUSSION

The modified algorithm was tested using CEC14 benchmark test functions, which include unimodal, simple multimodal, hybrid, and composition functions. Initial range, formulation, characteristics, and the dimensions of these problems are listed in Table I. High dimensions of the test functions is chosen, which is 50. To get a good analysis, the algorithm was run 51 times with 10000 iterations per trial. These setting parameters are shown in Table II. The standard BHA algorithm and the proposed BHLS algorithm for this study were coded in MATLAB2014 software on a Windows 7, 2.2 GHz, 6 GB RAM computer.

TABLE I. THE CEC 2014 BENCHMARK FUNCTIONS

Type of function	Function ID	The ideal value
Unimodal Functions	F1	100
	F2	200
	F3	300
Simple Multimodal Functions	F4	400
	F5	500
	F6	600
	F7	700
	F8	800
	F9	900
	F10	1000
	F11	1100
	F12	1200
	F13	1300
	F14	1400
	F15	1500
	F16	1600
Hybrid Functions	F17	1700
	F18	1800
	F19	1900
	F20	2000
	F21	2100
	F22	2200
Composition Functions	F23	2300
	F24	2400
	F25	2500
	F26	2600
	F27	2700
	F28	2800
	F29	2900
	F30	3000

The mean values produced by the tested algorithms are shown in Table III. The values shaded with green colour indicate the best performance of the proposed algorithm for each function.

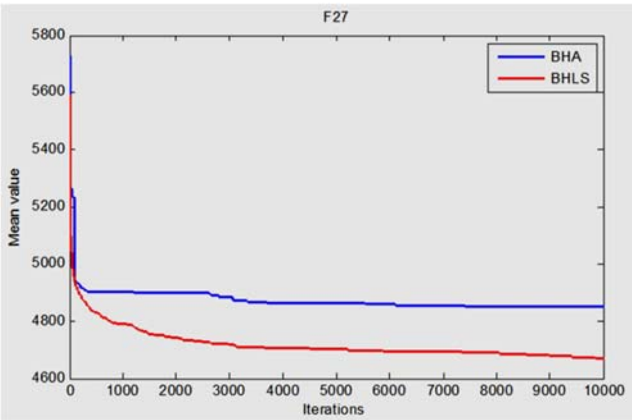

Figure 4. Convergence curve of Function 27.

TABLE II. THE SETTING PARAMETERS

Experimental parameters	
Iteration	10000
Number of trials	51
Number of agents	50
Number of dimensions	50
Search space	[-100 100]
rand	[0 1]

TABLE III. THE MEAN VALUE OF BHA AND BHLS

Functions	The ideal value	BHA	BHLS
F1	100	5611014.427	2552594.538
F2	200	4997329.195	8074.4764
F3	300	14041.12327	9527.208898
F4	400	609.8508916	517.3439441
F5	500	520.0161923	520.0010105
F6	600	658.7739009	659.6469821
F7	700	701.1662029	700.0109645
F8	800	953.499765	961.7926123
F9	900	1249.316564	1265.712813
F10	1000	3816.305871	3821.262884
F11	1100	8308.348714	8111.357044
F12	1200	1200.797984	1200.788607
F13	1300	1300.56279	1300.490845
F14	1400	1400.261316	1400.318211
F15	1500	1810.089933	1757.16024
F16	1600	1621.682464	1621.838854
F17	1700	639170.0635	249314.1797
F18	1800	2476.577727	4018.109097
F19	1900	1960.011302	1976.1636
F20	2000	9023.306146	5391.587955
F21	2100	429192.2267	198848.8719
F22	2200	3786.065395	3864.753976
F23	2300	2652.810511	2638.951606
F24	2400	2665.506218	2673.946081
F25	2500	2749.939581	2758.176312
F26	2600	2796.226359	2708.304169
F27	2700	4729.276014	4634.655362
F28	2800	11732.29457	11377.61756
F29	2900	10839.35982	34861.7663
F30	3000	69850.72279	18036.95308

The proposed BHLS algorithm able to provide good results for most of the tested functions when the results compared with the BHA. These results show the performance of the BHLS algorithm is more stable than BHA and it has good results in most of the tested functions, but for some functions, BHLS failed to outperform the BHA or to reach the optimum value as in F6, F8, F9, F10, F14, F16, F18, F19, F22, F24, F25, and F29. It can be seen from the results that BHLS generally worked as good as or sometimes better than BHA, especially for simple multimodal functions. BHLS tabulated good performance and able to outcome trapped in local optimum for most of these functions.

For hybrid functions, BHLS has good performance for some of these functions, in spite of the complexity, different variable and different. While in composition functions,

DOI 10.5013/IJSSST.a.17.04.12 ISSN: 1473-804x online, 1473-8031 print

BHLS outperform the BHA in some functions with good performance as in F26, F27, F28, and F30.

Fig. 4 shows an example of convergence curves of BHLS compared to BHA. These graphs show that the BHLS algorithm able to obtain better solution than the original BH algorithm at the end of the iteration.

VII. CONCLUSION

A local search method was employed to improve the BH algorithm. CEC2014 benchmark functions are employed for performance comparison. It is found that the proposed BHLS algorithm outperformed the standard BH algorithm for most of the functions. The next step is to apply the same local search to another new algorithm such as simulated Kalman filter [12-13].

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Dorigo and G. Di Caro, “Ant colony optimization: a new meta-heuristic,” in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, 1999, pp. 1470-1477.

[2] P.-W. Tsai, J.-S. Pan, B.-Y. Liao, and S.-C. Chu, “Enhanced artificial bee colony optimization,” International Journal of Innovative Computing, Information and Control, vol. 5, pp. 5081-5092, 2009.

[3] J. Kennedy, “Particle swarm optimization,” in Encyclopedia of machine learning, ed: Springer, 2011, pp. 760-766.

[4] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm,” Information sciences, vol. 179, pp. 2232-2248, 2009.

[5] A. Hatamlou, “Black hole: A new heuristic optimization approach for data clustering,” Information sciences, vol. 222, pp. 175-184, 2013.

[6] G. A. E.-N. A. Said, A. M. Mahmoud, and E.-S. M. El-Horbaty, “A comparative study of meta-heuristic algorithms for solving quadratic assignment problem,” arXiv preprint arXiv:1407.4863, 2014.

[7] A. Salhi and E. S. Fraga, “Nature-inspired optimisation approaches and the new plant propagation algorithm,” 2011.

[8] J. Liang, B. Qu, and P. Suganthan, “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, 2013.

[9] L. M. Gambardella and M. Dorigo, “Coupling ant colony system with local search,” 2015.

[10] E.-G. Talbi, Metaheuristics: from design to implementation, John Wiley & Sons, 2009.

[11] S. Ouadfel and M. Batouche, “Ant colony system with local search for Markov random field image segmentation,” in Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, 2003, pp. I-133.

[12] Z. Ibrahim, N. H. Abdul Aziz, N. A. Ab. Aziz, S. Razali, M. S. Mohamed, “Simulated Kalman Filter: a novel estimation-based metaheuristic optimization alogrithm,” Advance Science Letters, vol. 22, pp. 2941-2946, 2016.

[13] Z. Ibrahim, N. H. Abdul Aziz, N. A. Ab. Aziz, S. Razali, M. I. Shapiai, S. W. Nawawi, and M. S. Mohamed, “A Kalman filter approach for solving unimodal optimization problems,” ICIC Express Letters, vol. 9, issue 12, pp. 3415-3422, 2015.