

Article

3-D Terrain Node Coverage of Wireless Sensor Network Using Enhanced Black Hole Algorithm

Jeng-Shyang Pan ¹, Qing-Wei Chai ¹, Shu-Chuan Chu ^{1,*} and Ning Wu ²

¹ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; jspan@cc.kuas.edu.tw (J.-S.P.); mimanxiaowei@163.com (Q.-W.C.)

² School of Electronic and Information Engineering, Beibu Gulf University, Qinzhou 535011, China; n.wu@bbgu.edu.cn

* Correspondence: scchu0803@gmail.com

Received: 19 March 2020; Accepted: 21 April 2020; Published: 23 April 2020



Abstract: In this paper, a new intelligent computing algorithm named Enhanced Black Hole (EBH) is proposed to which the mutation operation and weight factor are applied. In EBH, several elites are taken as role models instead of only one in the original Black Hole (BH) algorithm. The performance of the EBH algorithm is verified by the CEC 2013 test suit, and shows better results than the original BH. In addition, the EBH and other celebrated algorithms can be used to solve node coverage problems of Wireless Sensor Network (WSN) in 3-D terrain with satisfactory performance.

Keywords: node coverage; 3-D node deployed; WSN; intelligence computing; black hole

1. Introduction

With the development of integrated circuits, micro-electromechanical systems (MEMS) and communication theory, the emergence of WSN has provided many conveniences for human life, and people can more conveniently monitor the surrounding environment or instruments easily. In WSN, optimizing the energy depletion ground on the limited size of sensor nodes while maintaining the data transmission rate is an enormous challenge. A self-adaptive clustering method is implemented to significantly save energy of sensor nodes [1]. Node coverage determines the detection property of WSN, and therefore achieving the maximum coverage rate with a given number of sensor nodes is of importance [2,3]. There are two methods to optimize sensor deployment: deterministic method and stochastic method. Deterministic method is used to deploy sensor nodes in a static environment for the predetermined positions of sensor nodes. The stochastic method randomly deploys sensor nodes with vehicles such as cars, planes, ships and et al. The deployment strategies for maximizing the coverage rate of WSN have not only been focused on the 2-D environment, but also on 3-D environment. Delaunay Triangulation and Voronoi Diagram have been used to find the sparse area of sensors and deploy new sensor nodes, which can significantly improve the coverage rate of WSN [4]. Bio-inspired algorithm has also been used to strengthen the capability of WSN [5], and PSO is applied to work out the energy-efficient coverage problem for the minimal cost [6]. Black Hole (BH) algorithm is an excellent meta-heuristic algorithm proposed in recent years [7]. It simulates the phenomenon that black holes engulf other planets in the universe. In the BH algorithm, the optimal solution is represented as the black hole and other candidate solutions are expressed as planets. The planet moves toward the black hole and finds the optimal solution on the way. If a planet is too close to a black hole, it will be engulfed by the black hole. Some simulations show that the BH algorithm has better performance in data clustering than some existing well-known algorithms.

Intelligent computing has been paid more and more attention by researchers because of its excellent performance in solving optimization problems, such as Artificial Bee Colony (ABC) [8–10],

Particle Swarm Optimization (PSO) [11–14], Genetic Algorithm (GA) [15], Ant Colony Optimization (ACO) [16–18], Cat Swarm Optimization (CSO) [19–21], Difference Evolution (DE) [22–24], Multi-Verse Optimizer [25,26], Symbiotic Organism Search Algorithm [27,28], QUATRE [29–31] and et al. To improve the optimization efficiency, many methods have been put forward. For example, a compact method is implemented to achieve better performance based on the memory of a single individual [32–37]. The simulation of the benchmark function can also effectively improve the speed and search-ability of the original algorithm [38]. In recent years, researchers are also inspired by the movement of animals to simulate the natural phenomena for optimization [39–42], and more and more improved algorithms have been proposed [43–46], including fuzzy processing [47–49].

Optimization methods are very important for increasing the efficiency of Wireless Sensor Network (WSN), which can efficiently monitor the object area at a low cost. In military applications, enemy invasion can be detected in real time by applying WSN. In agriculture applications, farmers can appropriately adjust the schedule or content of work after receiving the crop data obtained by the WSN. Moreover, we can predict the localization, velocity and travel time of vehicle according to the information of sensor nodes gathered [50,51].

This article presents a novel intelligence computing algorithm named Enhanced Black Hole (EBH) to which the mutation operation and weight factor are applied. The BH algorithm and 3-D node coverage problem of WSN are recommended in Section 2. Section 3 proposes an Enhanced version of the EH algorithm, and Section 4 apply the novel algorithm in solving 3-D node coverage problem of WSN. Section 5 shows the experiments and the results. Finally, in Section 6 a conclusion is drawn.

2. Black Hole Algorithm and 3-D Node Coverage Problem of WSN

2.1. Black Hole Algorithm

Black Hole (BH) algorithm is an excellent optimization algorithm for solving data clustering problems, inspired by the phenomenon about black hole devouring other stars [7]. In BH algorithm, the individual with the best fitness value is regarded as a black hole, and other individuals move towards it. If the distance between an individual and the black hole is smaller than the radius of event horizon R_{BH} , then it will be swallowed by the black hole and an individual will be randomly generated to maintain the size of the population. As other population-based algorithms, at the initial stage, candidate solutions are generated randomly in a finite region, and then updates the population to find the optimal value according to Equation (1).

$$Pos_i^{t+1} = Pos_i^t + rand \cdot (Pos_{BH}^t - Pos_i^t) \quad (1)$$

where the Pos_i^t means the location of i -th candidate solution at the t iteration and Pos_{BH}^t represents the black hole location at the t iteration. The $rand$ is a random number between 0 and 1. From Equation (1) we can learn about the BH algorithm gives up the impact of p_{best} , velocity, and two constants. It has been found that the BH algorithm outperforms the PSO algorithm in data clustering [7], this can prove it is a meaningful algorithm in some ways.

In BH algorithm, the event horizon is used to describe the range of being devoured by the black hole and it can be calculated by Equation (2) such that

$$R_{BH} = \frac{fit_{BH}}{\sum_{n=1}^N fit_n} \quad (2)$$

where R_{BH} is the radius of event horizon and fit_{BH} represents the best fitness value at the current iteration. The fitness value of n -th individual at the current iteration is represented by fit_n . The size of the population is represented by N . When the algorithm at an early stage, the population sparsely separates in the limited area. The black hole has the signally fitness value and the algorithm in the exploration phase. The longer the event horizon radius, the better to avoid the whole population being

concentrated in one place and the algorithm precocious. On the contrary, when the algorithm in the exploitation phase, the algorithm needs to further look for the optimal value around the promising area.

2.2. 3-D Node Coverage Problem of WSN

In this study, EBH algorithm is introduced to optimize the strategy of sensor nodes deployed in a 3-D terrain. Various factors will be taken into account when deploying sensors to a 3-D terrain than a 2-D terrain. When a sensor node communicates with other nodes, the transmission of the wireless signal may be interrupted, and the coverage rate of WSN will be affected.

In the simulation, sensor nodes are scattered randomly over a 3-D terrain generated by the “peak” function of MATLAB as shown in Figure 1, and some sensor nodes are placed at the intersection of the grid. The 3-D locations of nodes in this 3-D terrain are defined by the initial allocation, and the depth locations are determined by the following rules:

- If the node is at the grid intersection, the value of the third dimension of the location is the height of the three-dimensional terrain which showed in Figure 1.
- Otherwise, the value of the third dimension of the location is the height of the closest grid intersection.

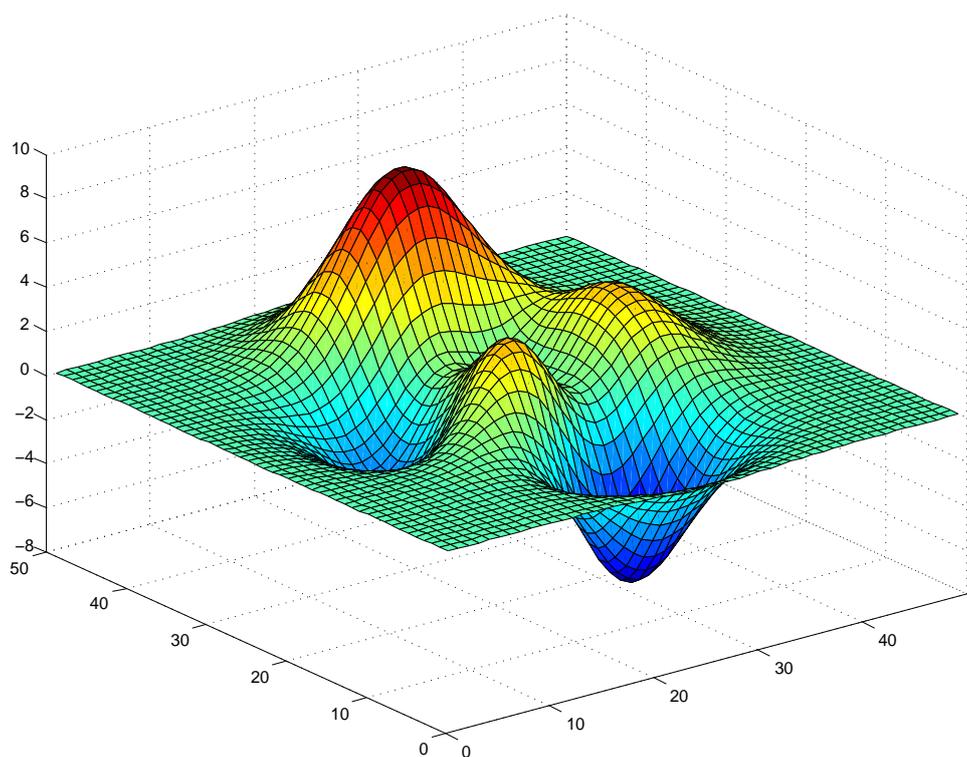


Figure 1. Terrain for Deploying Sensor Nodes (The units of x, y, z-axis is meter).

In the cause of study, the node coverage of WSN in this 3-D terrain, it is necessary to detect whether there is an obstacle between two nodes. The Bresenham Line of Sight (LOS) algorithm is adopted for the optimization of this study thanks to its computational efficiency. To determine whether the communication between the node and the target point is blocked, some points between S to P are selected, as shown in Figure 2. If the height of any point in the terrain is higher than the ray height of the same point, the communication between S and P will be blocked [52].

In WSN, the sensor nodes collect the information about object and transmits to the sink node for further processing. The sensing model is formulated to describe the coverage rate and it is affected

by the distance and other factors [53]. In this research, the binary sensing model is used to determine whether there is a connection between two nodes. Since this model only simulates the connectivity status of WSN, therefore it can only determine whether a sensor node is communicating with other nodes according to the communication radius. It can be presented by Equation (3), such that

$$C(r,s) = \begin{cases} 1, & \text{distance}(r,s) \leq R \text{ and there no obstacle} \\ 0, & \text{distance}(r,s) > R \text{ or there are obstacles} \end{cases} \quad (3)$$

where the r and s represent receive node and send node, respectively. The communication radius is represented by R . If the value of $C(r,s)$ is 1 means the node r can communicate with node s and vice versa.

In recent years, the Bresenham algorithm has been used in computer graphics to draw a line segment on a 2-D surface, in [53] the algorithm is modified and used it to determine LOS on a 3-D surface. Figure 2 shows a simplified paradigm about LOS on a 3-D terrain. It can be seen that if there is no pixel higher than the virtual line between node S and node P , these two nodes can communicate with each other, which means that there is a LOS between these two nodes.

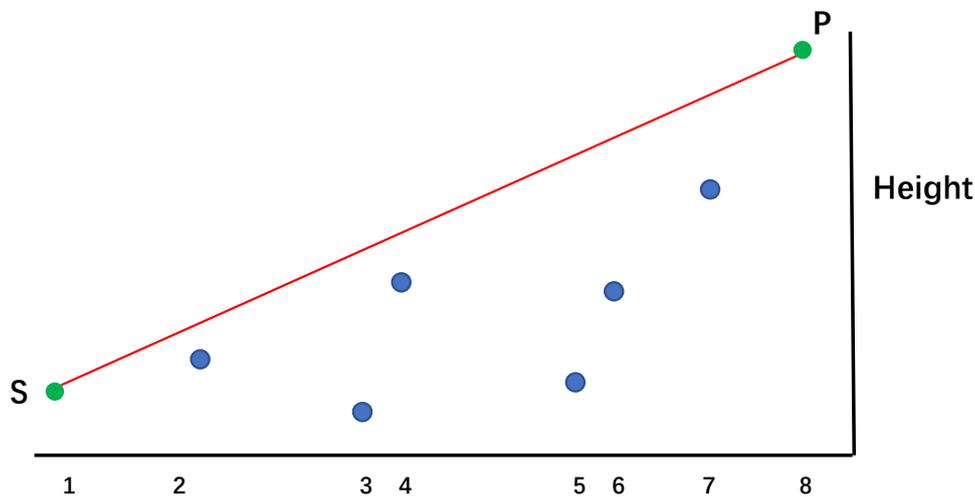


Figure 2. A Simple Paradigm about LOS.

3. Enhanced Black Hole Algorithm

BH algorithm is a competitive and novel algorithm, however its performance can be greatly improved. In BH algorithm, the motion of all individuals is only affected by the optimal solution position, and their trajectories are all straight lines. Therefore, there is not enough space for candidates to explore as they move towards the black hole. Although the BH algorithm has a rapid convergence rate and excellent performance in data clustering, it has a great chance to get stuck in the local optimal location. Therefore, a multiple black holes scheme has been considered, and the impact of the best candidate solution will be taken into account as well as the elite candidate solutions. In this study, we chose three best alternatives as models and applied varying degrees of gravity to other stars, such that

$$Pos_i^{t+1} = Pos_i^t + w \cdot [c_1 \cdot (Pos_{BH1}^t - Pos_i^t) + c_2 \cdot (Pos_{BH2}^t - Pos_i^t) + c_3 \cdot (Pos_{BH3}^t - Pos_i^t)] \quad (4)$$

where w represents the role of weight for each black hole, and c_1 , c_2 , and c_3 are three constants used to adjust the degree of influence of black holes. In the early stage, the individuals were sparsely distributed in the limited area, and the distances between the black holes and other stars are considerably longer than among the stars. Therefore, a heavier weight w can help to look for the optimal value in the

promising area quicker. If a promising area is found by the algorithm, the optimal value will be targeted within a smaller area instead of moving over a wider area. Therefore, a smaller value of w will be suitable for this situation. Mutation operation is added to the original BH algorithm, which enhances the global search ability of the algorithm. In each iteration, the values of the three random dimensions of the first black hole fluctuate 20% up and down based on their values. If the new position of the first black hole is better than the original position, it will replace the original position, thereby affecting the population to find the optimal solution. In general, the best candidate solution will have the biggest impact on other stars and other elite candidate solutions have a smaller influence on other stars. The pseudo-code of the novel algorithm is shown in Algorithm 1.

Algorithm 1: The Enhanced Black Hole Algorithm

```

Initialization:  $i = 1, t = 1, c_1 = 0.9, c_2 = 0.6, c_3 = 0.3, D = 20, n = 30, T = 1000$ ;
randomly initial the position of population and calculate the function value:  $Pos, fun$ ;
while  $t \leq T$  do
   $w = 2.0 - 1.6 \cdot \frac{t}{T}$ ;
  Rank the population by function value form small to large:
   $Pos = (Pos_1, Pos_2, \dots, Pos_n), fun = (fun_1, fun_2, \dots, fun_n)$ ;
   $Pos\_BH_1 = Pos_1, Pos\_BH_2 = Pos_2, Pos\_BH_3 = Pos_3, fun\_BH = fun_1$ ;
   $R\_BH = \frac{fun\_BH}{sum(fun)}$ ;
   $Pos\_mu = Pos\_BH_1$ ;
   $(d_1, d_2, d_3) \leftarrow$  randomly select form  $D$ ;
   $Pos\_mu(d_1, d_2, d_3) = (0.8 + 0.4 \cdot rand) \cdot Pos\_mu(d_1, d_2, d_3)$ ;
   $fun\_mu \leftarrow$  function( $Pos\_mu$ );
  if  $fun\_mu < fun\_BH$  then
     $Pos\_BH_1 = Pos\_mu$ ;
     $fun\_BH = fun\_mu$ ;
  end
  while  $i \leq n$  do
    Update  $Pos_i$  according to Equation (4) and  $fun_i$ ;
    if  $fun_i < fun\_BH$  then
       $Pos\_BH_1 = Pos_i$ ;
       $fun\_BH = fun_i$ ;
    end
    if The distance between  $Pos_i$  and  $Pos\_BH_1$  smaller than  $R\_BH$  then
      Initial  $Pos_i$  and compute  $fun_i$ 
    end
     $i = i + 1$ ;
  end
   $t = t + 1$ ;
end

```

4. Enhanced Black Hole Algorithm Applied on Node Coverage of WSN

In the EBH algorithm, each individual represents a deployment strategy of the node coverage problem. Since an individual represents the deployment positions of all nodes, the dimension of each individual is the product of the dimensions of the deployment environment and the number of nodes. However, the 3-D positions of the deployed terrain have been known, if the position of the 2-D plane is provided, then the value of the depth location can be calculated. Figure 3 illustrates the setting of values for each individual in the EBH algorithm.

$Node_{1,1}$	$Node_{1,2}$	$Node_{2,1}$	$Node_{2,2}$	$Node_{i,1}$	$Node_{i,2}$	$Node_{n,1}$	$Node_{n,2}$
--------------	--------------	--------------	--------------	-------	--------------	--------------	-------	--------------	--------------

Figure 3. The Value Setting of Dimensions of Individual.

Where the $Node_{i,1}$ and $Node_{i,2}$ are the first dimension and second dimension value of the i -th sensor node. The n represents the amount of the deployed sensor nodes. In the initial stage, each individual is randomly generated, which means that each deployment strategy randomly arranges sensor nodes. Except for the optimal strategy, all deployment strategies learn from the optimal strategy according to the updated rules of the EBH algorithm. Coverage Rate (Cr) is defined to evaluate the quality of deployment strategies and is given as,

$$Cr = \frac{1}{Q} \cdot \sum_{q=1}^Q \cdot \left(\sum_{p=1}^P C(s_p, r_q) \right) \quad (5)$$

where the Q denotes the number of pixels of the 3-D terrain and P is the number of sensor nodes. Through $C(s_p, r_q)$, we can know whether the q -th pixel of the terrain is covered by the p -th sensor node and value of $C(s_p, r_q)$ can be calculated by Equation (3).

5. Experiment Results and Discuss

An improved BH algorithm is proposed in this paper. To verify the optimal performance of the algorithm, it is compared with the original algorithm using the CEC 2013 test suite. Moreover, the novel algorithm is compared with the canonical algorithm PSO and the state-of-the-art algorithm WOA. In the CEC 2013, there are 28 benchmark functions which are denoted as $f1$ to $f28$ respectively. There are some unimodal functions are the first five functions, $f21$ to $f28$ are composition functions and others are basic multimodal functions.

All experiments in this article are performed on the same platform which is a notebook computer with an i5-7300HQ CPU @ 2.50GHz. The parameters of the algorithm have an important impact on the results. To fairly verify the performance of the algorithms participating in the comparison, the parameters that they jointly own are set to the same value. The unique parameter values for these algorithms are set according to the recommendations of the original [7,11,40]. Table 1 shows the setting of the algorithm parameters.

Table 1. Parameter Setting of Algorithms.

Algorithms	Common Parameters	Unique Parameters
Particle Swarm Optimization		$c = 2.0$, $w \in [0.4, 0.9]$, Velocity Range $\in [10, 10]$
Whale Optimization Algorithms	Population Size = 30 Iterations = 1000	$a \in [0, 2]$, $b = 1$
Black Hole	Dimensions = 20	NULL
Enhanced Black Hole	Limited Areas $\in [-100, 100]$	$w \in [0.4, 2.0]$, $c_1 = 0.9$, $c_2 = 0.6$, $c_3 = 0.3$

Table 2 shows the average and standard deviation of the results of the 30 runs of the four algorithms on the CEC 2013 test suite. The results obtained by EBH are superior to other algorithms on functions 6, 10, 21, 22, 25 and especially significantly improves the results on functions 1, 3 and 26. In terms of stability, the proposed algorithm has better performance than other algorithm on functions 1, 5, 6, 10, 24, 25 and 26. The PSO algorithm is well suited to optimize problems contained in CEC2013 and it shows the best performance in most test functions. In these test functions, the WOA algorithm and the original BH algorithm are not sufficiently competitive with other algorithms.

Table 2. Simulation Results of CEC 2013 Benchmark Function (The optimal value is marked by bold).

Functions	PSO		WOA		BH		EBH	
Variable	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f1	-1.36×10^3	1.58×10^3	-1.29×10^3	8.43×10^1	5.04×10^3	1.02×10^3	-1.40×10^3	9.51×10^{-7}
f2	1.22×10^6	7.61×10^5	4.69×10^7	2.25×10^7	2.84×10^7	4.47×10^6	2.92×10^6	1.62×10^6
f3	2.92×10^9	4.08×10^9	2.14×10^{10}	3.47×10^{10}	3.54×10^{15}	4.78×10^{15}	5.91×10^8	8.24×10^8
f4	5.55×10^3	3.47×10^3	8.46×10^4	2.99×10^4	4.39×10^4	1.20×10^4	2.96×10^4	8.27×10^3
f5	-9.68×10^2	5.26×10^1	-6.67×10^2	1.64×10^2	5.08×10^2	2.54×10^2	-9.69×10^2	2.73×10^1
f6	-8.54×10^2	3.73×10^1	-7.62×10^2	6.73×10^1	3.99×10^2	1.48×10^2	-8.61×10^2	3.71×10^1
f7	-7.18×10^2	4.16×10^1	1.18×10^4	6.04×10^4	3.84×10^4	3.25×10^4	-4.84×10^2	4.05×10^2
f8	-6.79×10^2	8.05×10^{-2}	-6.79×10^2	8.41×10^{-2}	-6.79×10^2	6.87×10^{-2}	-6.79×10^2	8.27×10^{-2}
f9	-5.80×10^2	2.87×10^0	-5.77×10^2	1.93×10^0	-5.77×10^2	2.29×10^0	-5.78×10^2	2.51×10^0
f10	-4.60×10^2	6.67×10^1	-2.95×10^2	7.51×10^1	1.97×10^2	7.37×10^1	-4.98×10^2	1.49×10^0
f11	-2.32×10^2	4.95×10^1	-8.97×10^1	7.55×10^1	-1.29×10^2	4.63×10^1	-2.16×10^2	4.90×10^1
f12	-1.39×10^2	4.80×10^1	-3.05×10^1	5.40×10^1	4.53×10^0	6.03×10^1	-1.01×10^2	5.57×10^1
f13	8.59×10^0	2.98×10^1	8.56×10^1	7.01×10^1	1.32×10^2	5.77×10^1	3.19×10^1	5.36×10^1
f14	2.35×10^3	4.36×10^2	3.45×10^3	4.64×10^2	4.03×10^3	5.65×10^2	2.76×10^3	5.77×10^2
f15	2.34×10^3	5.51×10^2	3.23×10^3	5.74×10^2	3.84×10^3	5.81×10^2	2.83×10^3	5.79×10^2
f16	2.01×10^2	3.85×10^{-1}	2.02×10^2	3.70×10^{-1}	2.01×10^2	4.66×10^{-1}	2.01×10^2	7.65×10^{-1}
f17	4.21×10^2	2.64×10^1	6.71×10^2	8.93×10^1	5.69×10^2	5.05×10^1	5.37×10^2	5.87×10^1
f18	5.14×10^2	2.44×10^1	7.69×10^2	5.75×10^1	6.85×10^2	5.94×10^1	6.20×10^2	6.39×10^1
f19	5.07×10^2	2.39×10^0	5.49×10^2	2.58×10^1	1.70×10^3	3.96×10^2	5.13×10^2	5.90×10^0
f20	6.10×10^2	2.38×10^{-1}	6.10×10^2	1.87×10^{-1}	6.10×10^2	1.63×10^{-1}	6.10×10^2	2.02×10^{-1}
f21	1.04×10^3	8.26×10^1	1.44×10^3	2.50×10^2	1.80×10^3	5.40×10^1	1.03×10^3	9.43×10^1
f22	4.49×10^3	5.69×10^2	4.78×10^3	5.28×10^2	5.52×10^3	4.74×10^2	4.15×10^3	7.89×10^2
f23	4.54×10^3	5.98×10^2	5.25×10^3	6.81×10^2	5.76×10^3	5.01×10^2	4.56×10^3	8.32×10^2
f24	1.26×10^3	1.85×10^1	1.27×10^3	6.50×10^0	1.29×10^3	8.80×10^0	1.27×10^3	8.77×10^0
f25	1.39×10^3	1.24×10^1	1.38×10^3	7.81×10^0	1.41×10^3	9.02×10^0	1.37×10^3	6.63×10^0
f26	1.51×10^3	6.65×10^1	1.47×10^3	8.01×10^1	1.41×10^3	2.08×10^0	1.40×10^3	4.42×10^{-1}
f27	2.16×10^3	7.73×10^1	2.26×10^3	5.24×10^1	2.33×10^3	6.09×10^1	2.19×10^3	6.32×10^1
f28	4.39×10^3	6.63×10^2	5.62×10^3	7.61×10^2	5.18×10^3	5.63×10^2	4.56×10^3	5.90×10^2

Through the analysis of the results in Table 2, the EBH algorithm does not have the best processing effect on multimodal problems. As described in the “there is no free lunch” theorem [54], “any performance improvement for one type of problem will be offset by performance for another type of problem.” EBH wants to get better on unimodal problems, the performance of the multimodal problem will sacrifice some of the global search ability.

Figures 4 and 5 show the process of optimization of four algorithms on 28 test functions. The abscissa is the number of iterations of the algorithm running, and the ordinate is the average value obtained from the optimization test function 30 times, a function value is recorded for every 100 iterations to make a graph to prove the search ability of the algorithm. The PSO, WOA, BH, and EBH algorithms are represented by “red”, “magenta”, “green” and “blue” respectively. In these figures, the EBH algorithm has faster convergence speed and stronger optimization ability than the original algorithm on unimodal problems, composition problems and most multimodal problems. Although the EBH algorithm has fast convergence speed, it is easily trapped at local optimal value on function 8 and 20. Therefore the original algorithm has better performance on these two functions.

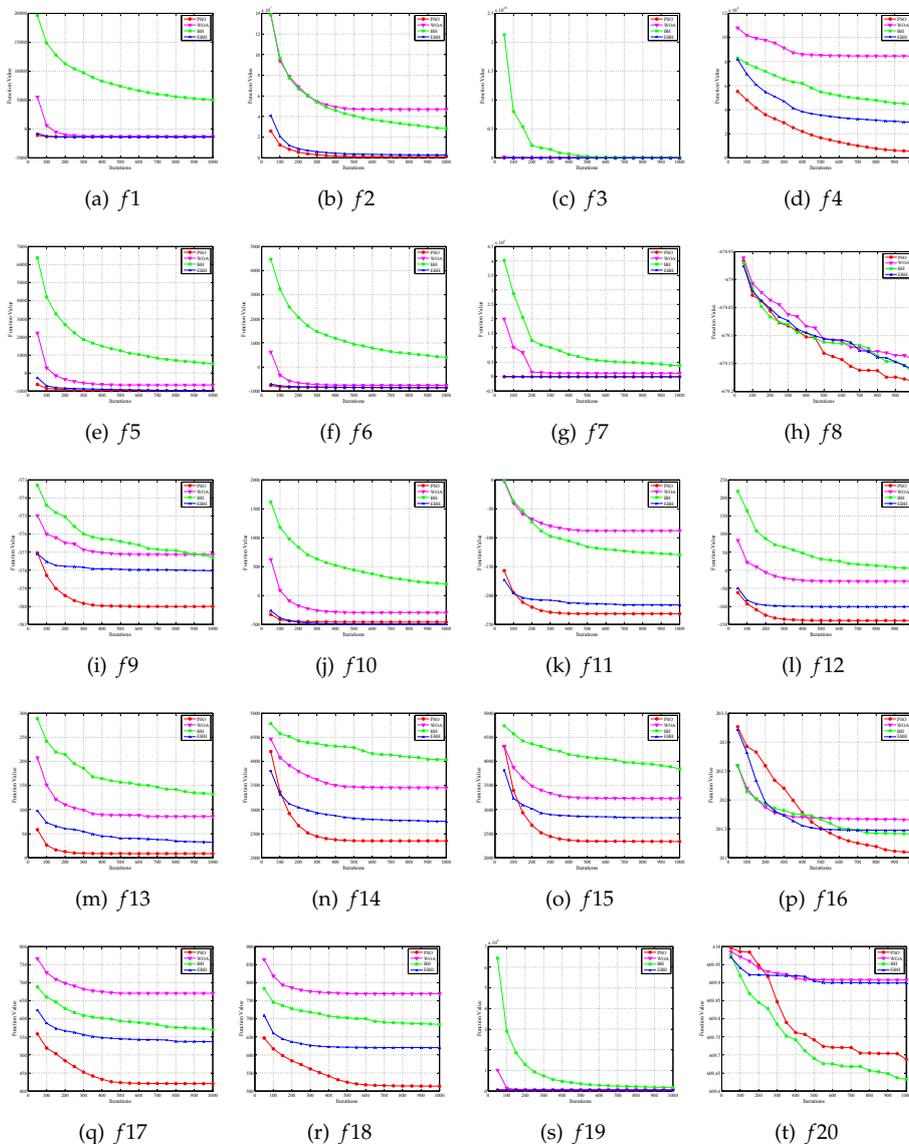


Figure 4. Results of Simulation Experiments (1).

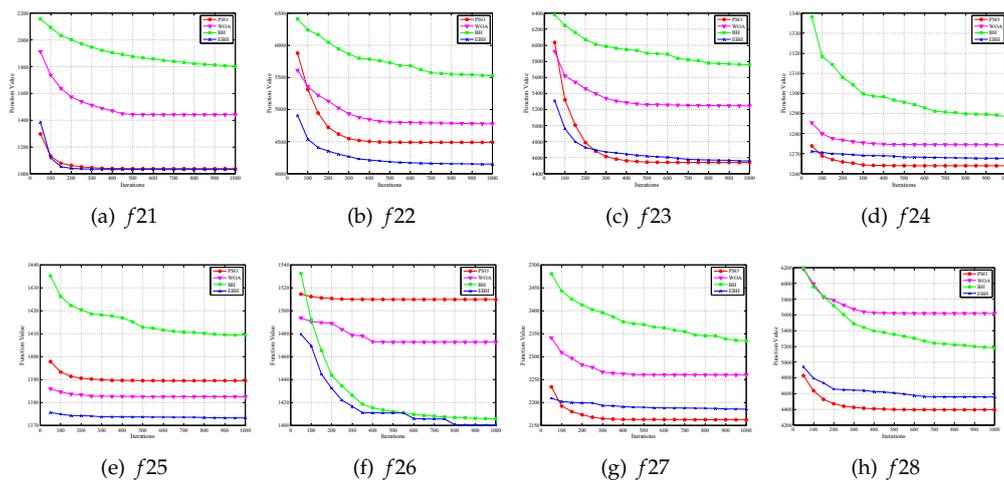


Figure 5. Results of Simulation Experiments (2).

In this article, the novel algorithm is used to solve the node coverage problem of WSN. This algorithm optimizes the positions of sensor nodes to achieve maximize coverage area with same account of nodes. The sensor nodes in the WSN are arranged in the terrain shown in Figure 1 and are tested in five groups of different number of nodes, which are 30, 40, 50, 60, and 70 nodes, respectively. In this simulation experiment, the area of plane of the terrain is 50×50 m and the communication radius is set to 5 m for each group.

Table 3 shows the results of 5 sets of simulations, which are the average of 30 runs. The data of different groups shows the influence of the number of nodes on the optimization results. When the communication radius is constant, the more nodes, the larger the area covered. Therefore, the difference between the optimization results of different algorithms decreases as the number of nodes increases. In Table 3, the new algorithm performs best on 30, 40, 50, and 70 sensor nodes. Although the difference is relatively small, in this optimization problem with few dimensions, we can still see the excellent performance of the new algorithm.

Table 3. Simulation Results of Node Coverage (The optimal value is marked by bold).

Functions	PSO	WOA	BH	EBH
30	45.55%	47.88%	47.82%	48.01%
40	55.53%	57.43%	57.75%	57.85%
50	62.88%	62.51%	64.99%	65.06%
60	69.44%	71.50%	71.32%	71.26%
70	74.71%	76.43%	76.43%	76.54%

6. Conclusions

In this paper, we enhanced the BH algorithm by considering more influencing factors and putting weight to the optimization of BH algorithm. Compared to the original BH algorithm, the EBH achieves a greater probability of falling into a promising region and avoiding local optimal values with a faster convergence speed. In the experiment with the CEC 2013 benchmark functions, the EBH algorithm shows better performance than the original BH in all test functions. The EBH algorithm has also been proved to be effective in solving the 3-D node coverage problems in WSN. The proposed EBH algorithm can also be applied to solve the cluster head problem of WSN in future.

Author Contributions: Conceptualization, J.-S.P.; Formal analysis, S.-C.C. and N.W.; Methodology, J.-S.P.; Software, Q.-W.C.; Supervision, S.-C.C.; Writing—original draft, Q.-W.C.; Writing—review and editing, N.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China(61872085), the Natural Science Foundation of Fujian Province (2018J01638) and the Fujian Provincial Department of Science and Technology (2018Y3001).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LD	Linear dichroism
MEMS	Micro-electro mechanical systems
PSO	Particle swarm optimization
ACO	Ant colony optimization
CSO	Cat swarm optimization
ABC	Artificial bee colony
GA	Genetic algorithm
DE	Difference evolution
QUATRE	Quasi-affine transformation evolution
WSN	Wireless sensor network
EBH	Enhanced black hole
BH	Black hole
LOS	Line of sight

References

1. Wang, J.; Gao, Y.; Wang, K.; Sangaiah, A.K.; Lim, S.J. An Affinity Propagation-Based Self-Adaptive Clustering Method for Wireless Sensor Networks. *Sensors* **2019**, *19*, 2579. [[CrossRef](#)] [[PubMed](#)]
2. Tsang, Y.P.; Choy, K.L.; Wu, C.H.; Ho, G.T.S. Multi-objective mapping method for 3D environmental sensor network deployment. *IEEE Commun. Lett.* **2019**, *23*, 1231–1235. [[CrossRef](#)]
3. Wang, J.; Ju, C.; Kim, H.J.; Sherratt, R.S.; Lee, S. A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs. *Clust. Comput.* **2019**, *22*, 1787–1795. [[CrossRef](#)]
4. Wu, C.H.; Lee, K.C.; Chung, Y.C. A Delaunay triangulation based method for wireless sensor network deployment. *Comput. Commun.* **2007**, *30*, 2744–2752. [[CrossRef](#)]
5. Kulkarni, R.V.; Venayagamoorthy, G.K. Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2010**, *40*, 663–675. [[CrossRef](#)]
6. Wang, J.; Ju, C.; Gao, Y.; Sangaiah, A.K.; Kim, G.J. A PSO based energy efficient coverage control algorithm for wireless sensor networks. *Comput. Mater. Contin.* **2018**, *56*, 433–446.
7. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [[CrossRef](#)]
8. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report, Technical Report-tr06; Erciyes University, Engineering Faculty, Computer: Kayseri, Turkey, 2005.
9. Tsai, P.W.; Pan, J.S.; Liao, B.Y.; Chu, S.C. Enhanced artificial bee colony optimization. *Int. J. Innov. Comput. Inf. Control* **2009**, *5*, 5081–5092.
10. Wang, H.; Wu, Z.; Rahnamayan, S.; Sun, H.; Liu, Y.; Pan, J.S. Multi-strategy ensemble artificial bee colony algorithm. *Inf. Sci.* **2014**, *279*, 587–603. [[CrossRef](#)]
11. Eberhart, R.; Kennedy, J. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Netw. Citeseer* **1995**, *4*, 1942–1948.
12. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [[CrossRef](#)]
13. Wang, H.; Wang, W.; Wu, Z. Particle swarm optimization with adaptive mutation for multimodal optimization. *Appl. Math. Comput.* **2013**, *221*, 296–305. [[CrossRef](#)]
14. Melin, P.; Olivas, F.; Castillo, O.; Valdez, F.; Soria, J.; Valdez, M. Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. *Expert Syst. Appl.* **2013**, *40*, 3196–3206. [[CrossRef](#)]
15. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [[CrossRef](#)]

16. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [[CrossRef](#)]
17. Chu, S.C.; Roddick, J.F.; Pan, J.S. Ant colony system with communication strategies. *Inf. Sci.* **2004**, *167*, 63–76. [[CrossRef](#)]
18. Chu, S.C.; Roddick, J.F.; Su, C.J.; Pan, J.S. Constrained ant colony optimization for data clustering. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, 26–30 August 2019; Springer: Berlin, Germany, 2004; pp. 534–543.
19. Chu, S.C.; Tsai, P.W.; Pan, J.S. Cat swarm optimization. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Guilin, China, 7–11 August 2006; Springer: Berlin, Germany, 2006; pp. 854–858.
20. Tsai, P.W.; Pan, J.S.; Chen, S.M.; Liao, B.Y.; Hao, S.P. Parallel cat swarm optimization. In Proceedings of the 2008 International Conference on Machine Learning and Cybernetics, Kunming, China, 5 September 2008; Volume 6, pp. 3328–3333.
21. Tsai, P.W.; Pan, J.S.; Chen, S.M.; Liao, B.Y. Enhanced parallel cat swarm optimization based on the Taguchi method. *Expert Syst. Appl.* **2012**, *39*, 6309–6319. [[CrossRef](#)]
22. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
23. Pan, J.S.; Liu, N.; Chu, S.C. A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning. *IEEE Access* **2020**, *8*, 17691–17712. [[CrossRef](#)]
24. Meng, Z.; Pan, J.S.; Tseng, K.K. PaDE: An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization. *Knowl. Based Syst.* **2019**, *168*, 80–99. [[CrossRef](#)]
25. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
26. Wang, X.; Pan, J.S.; Chu, S.C. A Parallel Multi-Verse Optimizer for Application in Multilevel Image Segmentation. *IEEE Access* **2020**, *8*, 32018–32030. [[CrossRef](#)]
27. Ezugwu, A.E.; Prayogo, D. Symbiotic Organisms Search Algorithm: Theory, recent advances and applications. *Expert Syst. Appl.* **2019**, *119*, 184–209. [[CrossRef](#)]
28. Chu, S.C.; Du, Z.G.; Pan, J.S. Symbiotic Organism Search Algorithm with Multi-Group Quantum-Behavior Communication Scheme Applied in Wireless Sensor Networks. *Appl. Sci.* **2020**, *10*, 930. [[CrossRef](#)]
29. Meng, Z.; Pan, J.S. QUasi-affine TRansformation Evolutionary (QUATRE) algorithm: A parameter-reduced differential evolution algorithm for optimization problems. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 4082–4089.
30. Meng, Z.; Pan, J.S. A competitive QUasi-Affine TRansformation Evolutionary (C-QUATRE) algorithm for global optimization. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Vancouver, BC, Canada, 24–29 July 2016; pp. 1644–1649.
31. Liu, N.; Pan, J.S.; Xue, J.Y. An Orthogonal QUasi-Affine TRansformation Evolution (O-QUATRE) Algorithm for Global Optimization. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing*; Springer: Berlin, Germany, 2020; pp. 57–66.
32. Harik, G.R.; Lobo, F.G.; Goldberg, D.E. The compact genetic algorithm. *IEEE Trans. Evol. Comput.* **1999**, *3*, 287–297. [[CrossRef](#)]
33. Dao, T.K.; Pan, J.S.; Nguyen, T.T.; Chu, S.C.; Shieh, C.S. Compact bat algorithm. In *Intelligent Data Analysis and Its Applications, Volume II*; Springer: Berlin, Germany, 2014; pp. 57–68.
34. Tian, A.Q.; Chu, S.C.; Pan, J.S.; Cui, H.; Zheng, W.M. A Compact Pigeon-Inspired Optimization for Maximum Short-Term Generation Mode in Cascade Hydroelectric Power Station. *Sustainability* **2020**, *12*, 767. [[CrossRef](#)]
35. Xue, X.; Chen, J. A Compact co-Firefly Algorithm for Matching Ontologies. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 2629–2632.
36. Xue, X.; Pan, J.S. A Compact Co-Evolutionary Algorithm for sensor ontology meta-matching. *Knowl. Inf. Syst.* **2018**, *56*, 335–353. [[CrossRef](#)]
37. Chu, S.C.; Xue, X.; Pan, J.S.; Wu, X. Optimizing ontology alignment in vector space. *J. Internet Technol.* **2020**, *21*, 15–22.
38. Sun, C.; Jin, Y.; Cheng, R.; Ding, J.; Zeng, J. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans. Evol. Comput.* **2017**, *21*, 644–660. [[CrossRef](#)]
39. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]

40. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
41. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**. [[CrossRef](#)]
42. Yang, X.S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin, Germany, 2010; pp. 65–74.
43. Hu, P.; Pan, J.S.; Chu, S.C.; Chai, Q.W.; Liu, T.; Li, Z.C. New Hybrid Algorithms for Prediction of Daily Load of Power Network. *Appl. Sci.* **2019**, *9*, 4514. [[CrossRef](#)]
44. Chai, Q.W.; Chu, S.C.; Pan, J.S.; Hu, P.; Zheng, W.M. A parallel WOA with two communication strategies applied in DV-Hop localization method. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–10. [[CrossRef](#)]
45. Pan, J.S.; Hu, P.; Chu, S.C. Novel Parallel Heterogeneous Meta-Heuristic and Its Communication Strategies for the Prediction of Wind Power. *Processes* **2019**, *7*, 845. [[CrossRef](#)]
46. Nguyen, T.T.; Pan, J.S.; Dao, T.K. An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network. *IEEE Access* **2019**, *7*, 75985–75998. [[CrossRef](#)]
47. Gustafson, D.E.; Kessel, W.C. Fuzzy clustering with a fuzzy covariance matrix. In Proceedings of the 1978 IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes, San Diego, CA, USA, 10–12 January 1979; pp. 761–766.
48. Chen, S.M.; Manalu, G.M.T.; Pan, J.S.; Liu, H.C. Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and particle swarm optimization techniques. *IEEE Trans. Cybern.* **2013**, *43*, 1102–1117. [[CrossRef](#)]
49. Chen, S.M.; Chang, Y.C.; Pan, J.S. Fuzzy rules interpolation for sparse fuzzy rule-based systems based on interval type-2 Gaussian fuzzy sets and genetic algorithms. *IEEE Trans. Fuzzy Syst.* **2012**, *21*, 412–425. [[CrossRef](#)]
50. Chen, C.H.; Lee, C.A.; Lo, C.C. Vehicle localization and velocity estimation based on mobile phone sensing. *IEEE Access* **2016**, *4*, 803–817. [[CrossRef](#)]
51. Chen, C.H.; Hwang, F.J.; Kung, H.Y. Travel time prediction system based on data clustering for waste collection vehicles. *IEICE Trans. Inf. Syst.* **2019**, *102*, 1374–1383. [[CrossRef](#)]
52. Topcuoglu, H.R.; Ermis, M.; Sifyan, M. Positioning and utilizing sensors on a 3-D terrain part I—Theory and modeling. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2010**, *41*, 376–382. [[CrossRef](#)]
53. Temel, S.; Unaldi, N.; Kaynak, O. On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *44*, 111–120. [[CrossRef](#)]
54. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).