An Improved Honey Badger Algorithm for Coverage Optimization in Wireless Sensor Network

Trong-The Nguyen[1,2,3], Thi-Kien Dao[1], Trinh-Dong Nguyen[2,3], Vinh-Tiep Nguyen[2,3]*

[1] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, China
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] VNUHCM, University of Information Technology, Ho Chi Minh City, Vietnam
thent@uit.edu.vn, jvnkien@gmail.com, {dongnt, tiepnv}@uit.edu.vn

Abstract

One of the critical indicators for a significant impact on the quality of service of wireless sensor networks (WSN) is coverage that directly determines the monitoring capability of the target monitoring area. This paper suggests a node coverage optimization strategy solution with an improved Honey-badger algorithm (called IHBA) to address the problems of WSN nodes' uneven distribution and low coverage in the random deployment. The honey-badger algorithm (HBA) is a recently developed metaheuristic algorithm with several advantages, e.g., robust process and ease of implementation; still, HBA has limitations in avoiding the local optimum trap when dealing with complicated node coverage optimization situations. The IHBA is implemented by modifying and updating equations with elite reverse learning and multi-direction strategies to prevent its original algorithm drawbacks. The obtained results on testing the benchmark function and the optimal WSN node coverage of the IHBA compared with the other algorithms in the literature. Compared results show that the IHBA algorithm provides effective optimal performance, convergence speed, and increasing feasible coverage. Significantly, the coverage rate archived of the IHBA is 87%, but the other methods are only below or equally 84% in the same comparison conditions.

Keywords: Coverage optimization, Improved honey badger algorithm, Wireless sensor network

1 Introduction

Wireless sensor networks (WSN) [1] consist of a collection of low-power sensor nodes with communication capability that has been widely employed in several areas, e.g., in the military, industrial, and agricultural control, urban management, and environmental monitoring applications [2]. The coverage problem is one of the most fundamental problems in WSNs, and coverage is an essential indicator for evaluating coverage optimization strategies. Coverage has a significant impact on WSN quality of service because it directly determines the monitoring capability of the target monitoring area. Sensor node deployment that is both rational and effective lowers network costs while also lowering energy usage [3]. WSN coverage applications attempt to deploy a small number of sensor nodes to monitor a specific target region of interest with coverage efficiency [4]. In most cases, sensor nodes are randomly placed in the target monitoring region, resulting in an uneven distribution of sensor nodes and low coverage [5]. As a result, it's critical to increase the node coverage of WSNs in the monitoring region by strategically placing sensor nodes. The rational and efficient deployment of WSN has been proven to be an NP-hard problem for large-scale sensor node deployment challenges, and finding the optimal solution for such situations remains a difficulty [6].

The metaheuristic algorithm is one of the promising ways considered as a remedy in this situation for dealing with the WSN nodes coverage [7]. Metaheuristic algorithms may identify near-optimal solutions in a fair amount of time with limited computational resources, making them a convenient approach to the WSN coverage optimization problem [8]. Metaheuristic algorithms are approximation optimization algorithms with solutions that effectively solve high-dimensional optimization problems [9]. Metaheuristic algorithms are often inspired by natural phenomena, e.g., human behaviors, physical sensations, animal swarm behaviors, evolutionary notions, and game theory [10].

Table 1 lists a typical example of metaheuristic algorithms according to the categories. Scholars have paid the potential application of the metaheuristic algorithms more attention to solving complicated nonlinear problems in engineering, final, and healthcare. Honeybadger algorithm (HBA) is a recently developed metaheuristic algorithm that mainly simulates the dynamic search behavior of honeybadger mining and searching honey. The limitation of the suggested method is that, like other metaheuristic methods, not all optimal cases guarantee obtained optimal results. In return, the proposed method gives acceptable and fast convergence optimal results. HBA has advantages, e.g., simple structure, few parameters, easy implementation, and would have broad application prospects in the future. Still, HBA has limitations in avoiding the local optimum trap when dealing with complicated node coverage optimization situations.

*Corresponding Author: Trong-The Nguyen; E-mail: thent@uit.edu.vn
DOI: 10.53106/160792642023032402015

Table 1. A typical example of meta-heuristic algorithms according to the categories

Categories based on	Algorithm's name and its symbol
Human-based	Exchange Market Algorithm (EMA) [11] Political Optimizer (PO) [12] Harmony search (HS) [13] Imperialist Competitive Algorithm (ICA) [14] Teaching Learning Based Optimization (TLBO) [15]
Animal-based	Particles Swarm Optimization (PSO) [16] Grey Wolf Optimizer (GWO) [17] Honeybadger Algorithm (HBA) [18] Salp Swarm Algorithm (SSA) [19] Dingo Optimization Algorithm (DOA) [20] African Vultures Optimization Algorithm (AVOA) [21], Northern Goshawk Optimization (NGO) [22] Reptile Search Algorithm (RSA) [23] Spotted Hyena-based Chimp Optimization (SSC) [24] Whale Optimization Algorithm (WOA) [25]
Physical phenomena	Simulated Annealing (SA) [26] Black Hole Algorithm (BH) [27] Sine Cosine Algorithm (SCA) [28] Ray Optimization (RO) [29]
Evolutionary concepts	Genetic Algorithm (GA) [30] Genetic Programming (GP) [31] Differential Evolution (DE) [32] Biogeography Based Optimizer (BBO) [33] Evolutionary Programming (EP) [34]
Game-based	Soccer League Competition (SLC) [35] League Championship Algorithm (LCA) [36] Volleyball Premier League (VPL) [37] Football Game-Based Optimization (FGO) [38] Ludo game-based metaheuristics (LGM)[39] Puzzle Optimization Algorithm (POA) [40]

This study suggests a solution to improving HBA (IHBA) to prevent its original algorithm drawbacks for the optimal WSN node coverage problem. The multiple-direction and elite reverse learning strategies are used to enhance diversity swarm agents and initialization in the IHBA implemented by modifying updating equations. The proposed IHBA algorithm is evaluated by testing the benchmark functions and nodes coverage problem and compared with the other selected popular algorithms in the literature to prove the potential performance of the algorithm.

The suggested approach contributions are highlighted as follows.

- Suggesting strategies for improving HBA to prevent its original algorithm drawbacks.
- Evaluating the suggested method's performance by testing the selected benchmark functions in CEC2017 and comparing the proposed method's results with the other algorithms in the literature.
- Establishing the suggested IHBA approach for the optimal WSN node coverage issue. Analyzing and discussing the results of the experiment.

2. Related Work

This section presents the WSN node coverage optimization as a statement model for the optimal problem and reviews the original Honeybadger optimization algorithm (HBO). The presentation subsections are detailed as follows.

2.1. WSN Node Coverage Model

The WSN node coverage optimization problem is the desired placement of each deployed node with a fixed sensing radius that each sensor can only perceive [41-42]. Each sensor can only sense and find within its sensing radius in monitoring desired deployment area [43].

Hence each node must be deployed subject to a constrained sensing radius in possible communication with each other and the entire network [44]. Its sensing radius is a suitable meet to the coverage issue of finding objects inside of it in possible optimization ranges [45].

Assuming WSN is set up in a two-dimensional monitoring region of $W{\cdot}L$ m^{2,} supposing that M sensor nodes are randomly deployed in that desired place. The rectangle area of the deployment network is divided into $W{\cdot}L$ grids of equal area size for ease of calculation, with the grid's center point being the monitoring node m. The number of sensors covering the complete monitoring area with minimum redundant sensing in an optimal condition is calculated as $W{\cdot}L{\cdot}M{\cdot}2R_S^2$ Let S be a set of nodes denoted as $S = \{S_i, i = 1, 2, ..., M\}$, and T be a set of target monitoring points, $T = \{T_j, j = 1, 2, ..., N\}$. Coordinates of for S_i and T_j are represented as $(S_i(x), S_i(y))$ and $(T_j(x), T_j(y))$ where $i = 1, 2, ..., M$ and $j = 1, 2, ..., N$. A sensor node sensing range is a circle with a center of sensing radius R_s as its radius [46-47].

The model of a two-dimensional WSN monitoring region network is assumed as follows:

- The sensing radius of each sensor node is R_s, and the communication radius is R_c, both in meters units with $R_c \geq 2R_s$.
- The sensor node can normally communicate function, have sufficient energy, and access time to data information.
- The sensor node has the same parameters, structure, and communication capabilities.
- The sensor node can move freely and update the location information in time.

WSN monitoring area. If T_j is covered by the sensor nodes, the distance between the target monitoring point T_j and any of the sensor nodes is less than or equal to the sensing radius Rs. Between sensor node S_i and goal monitoring point T_j, the Euclidean distance is defined as:

$$d\left(S_i, T_j\right) = \sqrt{\left(S_i(x) - T_j(x)\right)^2 + \left(S_i(y) - T_j(y)\right)^2}, \quad \textbf{(1)}$$

where $d(S_i, T_i)$ is the distance of node S_i to node T_j. The node sensing model is set on sensing radius if R_s is greater than or equal to $d(S_i, T_j)$, the probability p that the target is set to 1 otherwise it is set to 0. The probability formula is given as follows.

$$p(S_i, T_j) = \begin{cases} 1, & R_s \geq d(S_i, T_j) \\ 0, & R_s < d(S_i, T_j) \end{cases}, \quad \textbf{(2)}$$

The sensor nodes can work cooperatively by affecting neighbor nodes of the deployment two-dimensional WSN monitoring area. Whenever any target monitoring point can be covered by more than one sensor simultaneously, the probability of monitoring the target point reach sensed to T_j jointly as the formula given.

$$P(S,T_j) = -\prod_{i=1}^{M}(1-p(S_i,T_j)), \tag{3}$$

The coverage rate can be defined as the rate of the coverage area of all sensor nodes in the monitoring area to the total size of the monitoring area. It means that the coverage ratio calculation is the ratio of probability to the network deployed surface 2D WSN monitoring area as follows.

$$Cov_R = \frac{\sum_{j=1}^{N} P(S,T_j)}{W \cdot L}, \tag{4}$$

where Cov_R is the coverage ratio of WSN nodes, $P(S,T_j)$ is the probability of the target point reaching sensed node monitoring, and $W{\cdot}L$ is the 2D network deployed area [48].

2.2 Honeybadger Optimization Algorithm

The honeybadger algorithm (HBA) inspiration is taken from the kind of animal called the Badger in finding the prey as honey [18]. The dynamic search behavior for mining and searching for the honey of honeybadger is simulated for updated expression processing equations. The following are some of the characteristics of honeybadgers. Because of their fearless temperament will not hesitate to engage larger predators when they cannot flee circumstances. The honey badger employs the rat sniffing technique to wander slowly and repeatedly, searching for prey and excavating 50 holes in a radius of 40 kilometers or more per day.

Honeybadgers enjoy honey, but they aren't very good at locating beehives so they can work with birds. The bird takes the badger to the hives-honey and uses its long claws to help it open the hive, and they both get the benefits of teamwork. The HBA process model for the optimization algorithm includes population initialization, updating the search agent's location, prey attraction, and density factor [18].

Initialization with matric X solution is expressed as follows.

$$X = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} & \ldots\ldots & X_{1D} \\ X_{21} & X_{22} & X_{23} & X_{24} & \ldots\ldots & X_{2D} \\ X_{31} & X_{32} & X_{33} & X_{34} & \ldots\ldots & X_{3D} \\ X_{41} & X_{42} & X_{43} & X_{44} & \ldots\ldots & X_{4D} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots.. \\ X_{n1} & X_{n2} & X_{n3} & X_{n4} & \ldots\ldots & X_{nD} \end{pmatrix} \tag{5}$$

where X is a population composed of N honey badger individuals; X_j is the position of the jth individual among the N candidate individuals, D is dimension. It is randomly generated the initial position of a honey badger with the space search problem boundaries as follows:

$$X_{ij} = lb + r_1 \cdot (ub - lb), \tag{6}$$

where r_1 is a random number uniformly distributed on [0,1], lb and ub are the lower and upper bounds of the optimization space, respectively.

Updating the search agent's location has two stages: mining and enjoining honey stages. Before jumping into updating position expressions of the honey badgers mining and enjoining honey stages, some factors are considered, e.g., the attraction of prey and the density factor, to describe the expressions. The attraction of prey is related to the concentration intensity of prey and the distance between prey and the jth honey badger. K_j is the odor intensity of the prey (if the value of K_j is larger, it means that the jth honey badger can find the location of the prey more accurately and move to the prey faster), it is mathematically expressed as:

$$K_j = r_2 \cdot \frac{S}{4\pi d_j^2}, \tag{7}$$

where S and d_j are called source intensity (concentrated intensity) and distance between the prey and the jth honey badger. Two variables are expressed as follows.

$$S = (X_j - X_{j+1})^2, \tag{8}$$

$$d_j = X_p - X_j, \tag{9}$$

where X_P is the position of the prey regarded as the position of the optimal individual in the algorithm. The closer the honey badger is to its target, the stronger the attraction is.

The density factor decreases slowly with the number of iterations to ensure a smooth transition from exploration to development. The decreasing factor is updated with the over number of iterations to reduce randomization that is mathematically expressed as.

$$\alpha = C_0 \cdot \exp(\frac{-l}{l_{\max}}), \tag{10}$$

where, C_0 is a constant greater than or equal to 1, and the default is 2; l is the current number of iterations, and l_{max} is the maximum number of iterations.

The mining stage is the following expression gives the process of honey badgers looking for prey:

$$\begin{aligned} X_{\text{new}} &= X_p + F \cdot \beta \cdot K \cdot X_p \\ &\quad + F \cdot r_3 \cdot d_j \cdot \left|\cos(2\pi r_4)\left[1-\cos(2\pi r_5)\right]\right|, \end{aligned} \tag{11}$$

where X_p is the best location of prey; β greater than or equal to 1 that is the ability of honey badgers to obtain food; d_j is the distance between the prey and the *j-th* honey badger. r_3, r_4, and r_5 are three different random numbers between 0 and 1. K is prey odor intensity F is used as the search direction of the agent to change the search direction strictly.

$$F=\begin{cases}1, r_6\le 1/2\\ -1, r_6>1/2\end{cases}. \tag{12}$$

The honey stage is the second location update process with the honeyguide as birds have an inherent cooperative and mutually beneficial relationship. The honeyguide often looks for the hive in various places. Once it finds the hive's location, it will make a harsh scream. The update expression is given as follows.

$$X_{new}=X_p+F\cdot r_7\cdot\alpha\cdot d_j, \tag{13}$$

where X_{new} refers to the new location of the honey badger, while X_P refers to the location of the prey, F and α, and d_j are papermeter that are calculated in updating the honey badger searches at a position close to the X_P position.

3. Improved HBA Algorithm

This section presents strategies of elite reverse learning strategy for initialization and modifying the search direction to improve the HBA (namely, IHBA) algorithm. The performance and potential of the proposed method of the IHBA are verified in testing functions. The following subsections are in detail.

3.1 Algorithm Improvement

This subsection suggests elite reverse learning strategy for initialization, and modify the search direction to improve the performance optimization algorithm.

The initial population phase of the metaheuristic algorithm is an influential factor in the processing search for optimum performance. The reverse learning stagy obtains the population with the reverse solution according to the elite population, then integrates the population obtained by the reverse learning strategy with the initial population.

A reverse mechanism has a good effect on increasing population diversity and improving population quality [49]. Generating reverse solutions can effectively improve the diversity of solutions and be closer to the optimal solution. It selects high-quality individuals with the same number as the initial population to form a new initial population. The reverse learning equation is defined as follows:

$$X'_m=r\cdot(u+e)-X_m, \tag{14}$$

where r is the random value on [0,1], X_m is the position of the current individual, u and e represent the upper and lower bounds of the problem search space which means the X_m is in [u, e]. Then select excellent individuals from the reverse and current populations to form a new population applied initialization with X_j of the Eq.(6).

A multi-directing strategy: The direction of motion F and its expression in Eq.(8) has just two motion directions; however, the complicated problem's space may have more scales reaching motions space. A random permutation integer is generated without repeating elements chosen at random from the integers for creating the different searching directions. The motion direction F is alternated formula expression as follows.

$$F_{new}=\begin{cases}+1\cdot rand, \ if\ r_6\le 0.5\\ -1\cdot rand, \ otherwise\end{cases}. \tag{15}$$

where F_{new} an alternated direction guiding factor; *rand* is a random number with a range from 0 to 1 that helps to create the different searching directions.

The strategy of multi-directing F_{new} is hybridized into updated formulas for generating new solutions. An updating of the position of the honey badger is conducted as follows.

$$X_{new}=\begin{cases}X_p+F\cdot\beta\cdot K\cdot X_p+F\cdot r_3\cdot d_j\cdot\left|\cos(2\pi r_4)[1-\cos(2\pi r_5)]\right|, r_2\le 0.5\\ X_p+F_{new}\cdot r_7\cdot\alpha\cdot d_j, r_2>0.5\end{cases}, \tag{16}$$

where X_{new} is the updated position of the honey badger; d_j is the distance between the jth honey badger and the prey. The flow chart of the algorithm is shown in Figure 1.

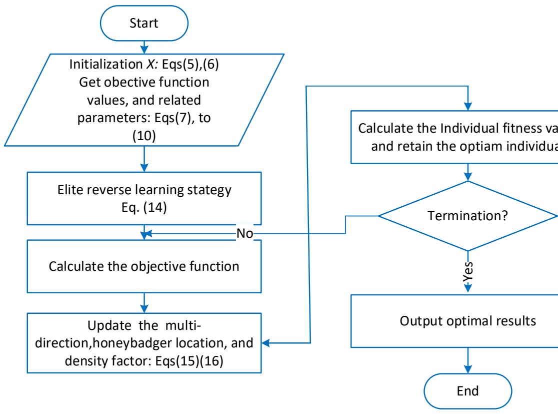


Figure 1. Flow chart of the IHBA approach

3.2 Experimental Results on Mathematic Test Functions

The potential better performance of the proposed IHBA algorithm needs to be proven by testing the benchmark function and compared with the other selected popular algorithms in the literature. The selected benchmark functions of the CEC2017 test suit are used in the practical experiment. The selected popular algorithms, e.g., the original HBA, PSO [50], GWO [17], MFO [51], DOA [20], AVOA [21], NGO [22], RSA [23], SSC [24], and WOA [25] are used in experimenting with the same conditional setting environment to compare with the proposed IHBA algorithm. The number runs times are set to 20 for each test function independently for obtaining the outcome results to ensure the fairness and accuracy of the investigation. See Table 2 shows for specific parameter settings of these algorithms.

Table 2. Parameter setting of the algorithms

Algorithm	Parameter settings
IHBA	Pop=40, iteration=1000, $C = 2$, $\beta = 6$
HBA [18]	Pop=40, iteration=1000, $C = 2$, $\beta = 6$
PSO [50]	Pop=40, iteration=1000, $\omega = 0.9$ to 0.4, $V_{max} = 10$, $V_{min} = -10$, $c_1 = 1.564$, $c_2 = 1.564$
MFO [51]	Pop=40, iteration=1000, $a = -1$, $b = 1$
GWO [17]	Pop=40, iteration=1000, $\alpha \in [0, 2]$
DOA [20]	Pop=40, iteration=1000, $P = 0.5$, $Q = 0.7$ 7
AVOA [21]	$u,v \in [0, 1]$, iteration=1000, $\beta = 1.5$
NGO [22]	$r \in [0, 1]$, $I \in [1, 2]$ iteration=1000
RSA [23]	$\alpha = 0.1$, $\beta = 0.2$ iteration=1000, $r_3 \in [-1, 1]$
SSC [24]	$r_1, r_2 \in [0, 1]$, iteration=1000, $l \in [2.5$ to $0]$
WOA [25]	$a \in [2, 0]$, $\vec{A} \in [-1, 1]$, iteration=1000

Tables 3 to Table 5 show paired comparisons of the optimal values of different algorithms with the IHBA under the test functions with 50D. In the Table, the smaller the value corresponding to the algorithm, the better the performance. The summarised in the last of Tables are terms of 'lose,' 'win,' or 'draw' mean the numbers of worse, better, and the same performance of the proposed IHBA algorithm.

According to Table 3, from the optimal value of BEST, the IHBA algorithm has 21 better, 5 similar and 1 worse performance than HBA algorithm respectively from the optimal value of MEAN, it has 20 better, 6 similar and 2 worse performances respectively from the perspective of standard deviation-STD., it has 19 better, 9 similar and 0 worse performances respectively. The same analysis comparison with the FMO, PSO, and GWO algorithms is summarized in rows in the last Tables showing that the proposed IHBA approach produces a better performance than the other competitors.

Table 5 shows the paired further comparison of optimization performance of the IHBA with five other algorithms, e.g., AVOA, NGO, RSA, SSC, and WOA, on 28 test functions with the optimal value of MEAN. It can be seen that the IHBA algorithm has many 'Win' higher than the worse and similar performances. The paired comparison of the algorithms is summarized in rows in the last Tables showing that the proposed IHBA approach performs better than the other compared algorithms.

Table 3. Paired comparison of optimization performance of the IHBA with HBA, and MFO on 28 classical test functions

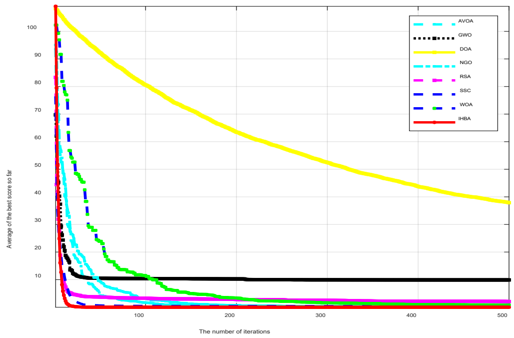
50D	HBA			MFO			IHBA		
	BEST	MEAN	STD.	BEST	MEAN	STD.	BEST	MEAN	STD.
CEC1	4.275E-11	2.819E-08	3.896E-08	6.281E+03	2.007E+04	1.287E+04	**9.095E-13**	**3.251E-12**	**3.101E-12**
CEC2	1.410E+06	2.891E+06	1.097E+06	2.922E+07	8.910E+07	3.405E+07	**1.012E+06**	**1.780E+06**	**4.896E+05**
CEC3	8.472E+08	2.885E+09	1.897E+09	9.222E+10	1.315E+11	4.089E+10	**9.418E+07**	**6.607E+08**	**5.651E+08**
CEC4	2.309E+04	3.001E+04	3.832E+03	9.613E+04	1.545E+05	3.981E+04	**8.116E-02**	**2.467E+03**	**9.795E-02**
CEC5	4.304E-09	7.828E-08	9.908E-08	3.943E+03	6.953E+03	3.253E+03	**3.592E-09**	**9.805E-09**	**8.290E-09**
CEC6	4.345E-01	7.022E-01	3.537E-01	3.101E-02	1.278E+03	8.130E-02	**4.345E-02**	**5.619E-02**	**2.207E-02**
CEC7	5.806E-01	8.335E-01	**1.357E-01**	1.586E-02	2.320E-02	3.592E-01	**5.454E-01**	**7.741E-01**	1.384E-01
CEC8	**2.109E-01**	**2.118E-01**	5.054E-02	2.108E-01	2.118E-01	5.639E-02	2.112E-01	2.119E-01	**3.933E-02**
CEC9	**4.456E-01**	5.445E-01	6.338E+00	5.198E-01	5.992E-01	**5.456E+00**	4.316E-01	**5.062E-01**	6.212E+00
CEC10	1.351E-01	8.370E-01	4.430E-01	1.462E+03	3.779E+03	1.256E+03	**3.334E-02**	**7.884E-02**	**2.748E-02**
CEC11	1.333E-02	1.969E-02	4.762E-01	3.176E-02	4.874E-02	1.791E-02	**1.184E-02**	**1.856E-02**	**3.436E-02**
CEC12	2.000E-02	2.858E-02	7.266E-01	6.369E-02	7.626E-02	1.130E-02	**1.711E-02**	**2.306E-02**	**3.647E-01**
CEC13	3.982E-02	5.157E-02	8.632E-01	5.894E-02	8.308E-02	1.627E-02	**2.960E-02**	**4.325E-02**	**8.045E-01**
CEC14	3.903E+03	**5.667E+03**	1.421E+03	4.491E+03	6.968E+03	**1.218E+03**	**3.886E+03**	6.194E+03	1.982E+03
CEC15	7.019E+03	1.007E+04	2.940E+03	7.921E+03	9.458E+03	**1.213E+03**	**6.157E+03**	**8.634E+03**	2.004E+03
CEC16	2.799E+00	3.585E+00	4.015E-01	**1.192E+00**	**1.856E+00**	**8.336E-01**	1.289E+00	3.145E+00	1.149E+00
CEC17	**1.701E-02**	**2.830E-02**	5.091E-01	3.084E-02	8.274E-02	5.589E-02	2.008E-02	2.917E-02	**4.512E-01**
CEC18	2.805E-02	**3.725E-02**	**7.485E-01**	4.577E-02	1.166E+03	4.402E-02	**2.308E-02**	3.785E-02	1.336E-02
CEC19	1.355E-01	2.954E-01	1.233E-01	1.635E+04	4.646E+05	3.379E+05	**9.806E+00**	**1.640E-01**	**4.252E+00**
CEC20	2.146E-01	2.244E-01	8.413E-01	2.176E-01	2.381E-01	**9.318E-01**	**1.885E-01**	**2.194E-01**	1.635E+00
CEC21	**2.000E-02**	**8.871E-02**	**2.801E-02**	1.128E+03	2.130E+03	8.655E-02	**2.000E-02**	9.443E-02	2.942E-02
CEC22	5.129E+03	7.234E+03	1.901E+03	5.004E+03	7.317E+03	1.243E+03	**4.887E+03**	**6.586E+03**	**1.133E+03**
CEC23	**7.479E+03**	1.095E+04	1.953E+03	9.269E+03	**1.061E+04**	**9.228E-02**	7.571E+03	1.103E+04	2.111E+03
CEC24	3.958E-02	4.306E-02	2.603E-01	**3.206E-02**	**3.559E-02**	**1.733E-01**	3.433E-02	3.860E-02	2.874E-01
CEC25	4.040E-02	4.530E-02	2.168E-01	**3.570E-02**	**3.746E-02**	**1.194E-01**	3.769E-02	4.097E-02	2.247E-01
CEC26	4.032E-02	4.705E-02	3.343E-01	**2.064E-02**	**4.296E-02**	7.885E-01	4.097E-02	4.465E-02	**1.801E-01**
CEC27	1.619E+03	2.065E+03	2.208E-02	1.691E+03	1.884E+03	**1.189E-02**	**1.599E+03**	**1.872E+03**	1.828E-02
CEC28	**4.000E-02**	1.537E+03	1.833E+03	1.594E+03	4.518E+03	1.874E+03	**4.000E-02**	**1.417E+03**	**1.639E+03**
Win	21	20	19	22	22	18	---	---	---
Lose	5	6	9	6	6	10	---	---	---
Draw	2	2	0	0	0	0	---	---	---

Table 4. Paired comparison of optimization performance of the IHBA with PSO, GWO on 28 classical test functions

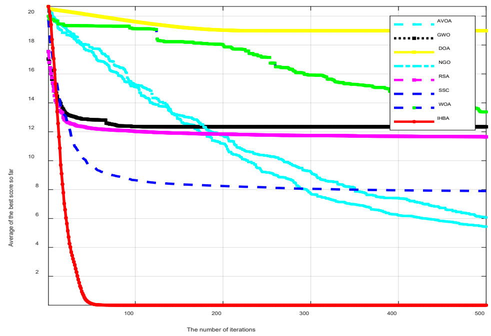
50D	PSO			GWO			IHBA		
	BEST	MEAN	STD.	BEST	MEAN	STD.	BEST	MEAN	STD.
CEC1	1.286E+03	3.283E+03	1.233E+03	3.234E+04	4.544E+04	9.092E+03	**9.095E-13**	**3.251E-12**	**3.101E-12**
CEC2	1.403E+07	4.175E+07	2.897E+07	1.594E+08	2.893E+08	1.222E+08	**1.012E+06**	**1.780E+06**	**4.896E+05**
CEC3	1.034E+10	1.848E+10	6.463E+09	5.265E+10	4.629E+12	1.399E+13	**9.418E+07**	**6.607E+08**	**5.651E+08**
CEC4	4.355E+04	5.316E+04	7.028E+03	4.197E+04	6.288E+04	1.269E+04	**8.116E-02**	**2.467E+03**	**9.795E-02**
CEC5	5.980E-02	9.207E-02	2.726E-02	3.557E+03	5.850E+03	2.279E+03	**3.592E-09**	**9.805E-09**	**8.290E-09**
CEC6	1.631E-02	2.382E-02	6.161E-01	3.557E+03	5.850E+03	2.279E+03	**4.345E-01**	**5.619E-01**	**2.207E-01**
CEC7	**4.201E-01**	**6.216E-01**	1.819E-01	1.416E-02	1.986E-02	5.511E-01	5.454E-01	7.741E-01	**1.384E-01**
CEC8	**2.106E-01**	2.120E-01	5.100E-02	2.114E-01	2.124E-01	4.062E-02	2.112E-01	**2.119E-01**	**3.933E-02**
CEC9	**3.853E-01**	**4.035E-01**	**2.534E+00**	5.635E-01	6.413E-01	5.538E+00	4.316E-01	5.062E-01	6.212E+00
CEC10	4.013E-02	5.396E-02	7.968E-01	2.271E+03	5.127E+03	1.555E+03	**3.334E-02**	**7.884E-02**	**2.748E-02**
CEC11	1.664E-02	2.199E-02	2.987E-01	6.063E-02	7.734E-02	1.225E-02	**1.184E-02**	**1.856E-02**	**3.436E-01**
CEC12	1.904E-02	2.455E-02	3.600E-01	6.556E-02	7.685E-02	9.322E-01	**1.711E-02**	**2.306E-02**	**3.647E-01**
CEC13	3.158E-02	**4.003E-02**	**6.643E-01**	5.665E-02	8.107E-02	1.282E-02	**2.960E-02**	4.325E-02	8.045E-01
CEC14	5.536E+03	7.168E+03	2.194E+03	1.096E+04	1.310E+04	**1.584E+03**	**3.886E+03**	**6.194E+03**	1.982E+03
CEC15	6.465E+03	9.668E+03	3.462E+03	1.105E+04	1.410E+04	**1.506E+03**	**6.157E+03**	**8.634E+03**	2.004E+03
CEC16	3.471E+00	**3.761E+00**	**1.965E-01**	3.324E+00	4.113E+00	5.136E-01	**1.289E+00**	3.145E+00	1.149E+00
CEC17	2.558E-02	3.516E-02	8.566E-01	7.530E-02	1.125E+03	1.643E-02	**2.008E-02**	**2.917E-02**	**4.512E-01**
CEC18	4.581E-02	5.639E-02	**5.779E-01**	1.075E+03	1.228E+03	1.018E-02	**2.308E-02**	**3.785E-02**	1.336E-02
CEC19	3.161E-01	2.938E-02	2.735E-02	1.608E+04	5.211E+04	3.763E+04	**9.806E+00**	**1.640E-01**	**4.252E+00**
CEC20	1.987E-01	**2.132E-01**	8.053E-01	2.349E-01	2.426E-01	**4.237E-01**	**1.885E-01**	2.194E-01	1.635E+00
CEC21	1.150E+03	2.277E+03	8.099E-02	3.758E+03	4.058E+03	**1.563E-02**	**2.000E-02**	**9.443E-02**	2.942E-02
CEC22	5.960E+03	6.949E+03	**6.839E-02**	1.071E+04	1.314E+04	1.597E+03	**4.887E+03**	**6.586E+03**	1.133E+03
CEC23	**6.253E+03**	8.859E+03	2.783E+03	1.088E+04	1.323E+04	**1.518E+03**	7.571E+03	**1.103E+04**	2.111E+03
CEC24	**2.803E-02**	**3.087E-02**	1.726E-01	3.841E-02	4.056E-02	**1.539E-01**	3.433E-02	3.860E-02	2.874E-01
CEC25	**3.303E-02**	3.471E-02	**1.097E-01**	4.028E-02	4.275E-02	**2.287E-01**	3.769E-02	4.097E-02	2.247E-01
CEC26	3.834E-02	**3.982E-02**	**8.582E+00**	**2.060E-02**	4.211E-02	1.066E-02	4.097E-02	4.465E-02	1.801E-01
CEC27	**1.128E+03**	**1.396E+03**	1.394E-02	2.033E+03	2.184E+03	**1.101E-02**	1.599E+03	1.872E+03	1.828E-02
CEC28	7.023E-02	**1.058E+03**	**3.632E-02**	5.871E+03	7.332E+03	7.978E-02	**4.000E-02**	1.417E+03	1.639E+03
Win	20	18	18	27	27	16	---	---	---
Lose	8	10	10	1	1	12	---	---	---
Draw	0	0	0		0	0	---	---	---

Table 5. Paired comparison of optimization performance of the IHBA with AVOA, NGO, RSA, SSC, and WOA on 28 test functions

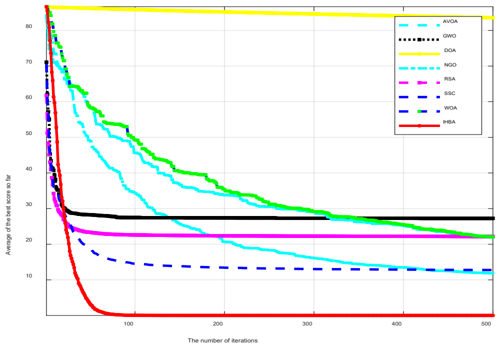
50D	AVOA	NGO	RSA	SSC	WOA	IHBA
CEC1	2.01E-01(>)	2.82E-08(>)	3.75E-01(>)	1.05E-11(>)	3.72E-12(<)	4.25E-12
CEC2	8.92E+02(>)	2.89E+02(>)	8.91E+02(>)	1.16E+03(<)	5.88E+02(>)	1.78E+02
CEC3	1.42E+07(>)	2.89E+07(>)	1.32E+07(>)	1.08E+06(>)	6.78E+05(>)	6.61E+02
CEC4	2.55E+03(<)	3.00E+04(>)	1.55E+03(<)	9.33E-02(<)	1.18E-01(<)	2.47E+03
CEC5	6.95E+03(>)	7.83E-08(>)	6.95E+03(>)	4.13E-09(<)	9.95E-09(>)	9.81E-09
CEC6	1.28E+03(>)	7.02E-01(>)	1.28E+03(>)	5.00E-02(<)	5.65E-02(~)	5.62E-02
CEC7	2.32E-02(<)	8.34E-01(>)	2.32E-02(>)	6.27E-01(<)	1.66E-01(<)	7.74E-01
CEC8	2.12E-01(~)	2.12E-01(~)	2.12E-01(~)	2.43E-01(>)	4.72E-02(>)	2.12E-01
CEC9	5.99E-01(>)	5.45E-01(~)	5.99E-01(>)	4.96E-01(<)	7.45E+00(>)	5.06E-01
CEC10	3.78E+03(>)	8.37E-01(>)	3.78E+03(>)	3.83E-02(<)	3.30E-02(<)	7.88E-02
CEC11	4.87E-02(>)	1.97E-02(>)	4.87E-02(>)	1.86E-02(~)	4.12E-02(>)	1.86E-02
CEC12	7.63E-02(>)	2.86E-02(>)	7.63E-02(>)	2.97E-02(>)	4.38E-01(>)	2.31E-02
CEC13	8.31E-02(>)	5.16E-02(>)	8.31E-02(>)	3.40E-02(<)	4.35E-02(~)	4.33E-02
CEC14	6.97E+03(>)	5.67E+03(<)	6.97E+03(>)	4.47E+03(<)	2.38E+03(<)	6.19E+03
CEC15	9.46E+03(>)	1.01E+04(<)	9.46E+03(>)	7.08E+03(<)	2.40E+03(<)	8.63E+03
CEC16	1.86E+00(<)	3.59E+00(>)	1.86E+00(<)	1.48E+00(<)	1.38E+00(<)	3.15E+00
CEC17	8.27E-02(>)	2.83E-02(>)	2.87E-02(<)	2.31E-02(<)	5.41E-01(>)	2.92E-02
CEC18	1.17E+03(>)	3.79E-02(~)	1.17E+03(>)	2.65E-02(<)	1.60E-02(<)	3.79E-02
CEC19	4.65E+05(>)	2.95E-01(>)	4.65E+05(>)	1.13E+01(>)	5.10E+00(>)	1.64E-01
CEC20	2.38E-01(>)	2.24E-01(>)	2.38E-01(>)	2.17E-01(<)	1.96E+00(>)	2.19E-01
CEC21	2.13E+03(>)	8.87E-02(<)	2.13E+03(>)	2.30E-02(<)	3.53E-02(<)	9.44E-02
CEC22	7.32E+03(>)	7.23E+03(>)	7.32E+03(>)	5.62E+03(<)	1.36E+03(<)	6.59E+03
CEC23	1.06E+04(>)	1.10E+04(~)	1.06E+04(<)	8.71E+03(>)	2.53E+03(<)	1.10E+04
CEC24	3.56E-02(<)	4.31E-02(<)	3.56E-02(>)	3.95E-02(>)	3.45E-01(>)	3.86E-02
CEC25	3.75E-02(>)	4.53E-02(>)	3.75E-02(<)	4.13E-02(~)	2.70E-01(<)	4.10E-02
CEC26	4.48E-02(~)	4.71E-02(~)	4.30E-02(<)	4.71E-02(>)	2.16E-01(<)	4.47E-02
CEC27	1.88E+03(~)	2.07E+03(<)	1.88E+03(~)	1.88E+03(~)	2.19E+02(>)	1.87E+02
CEC28	4.52E+03(>)	1.54E+03(>)	4.52E+03(>)	4.60E-02(<)	1.47E+03(~)	1.42E+03
Win	21	19	20	15	16	--
Lose	4	4	6	10	9	--
Draw	3	5	2	3	3	--



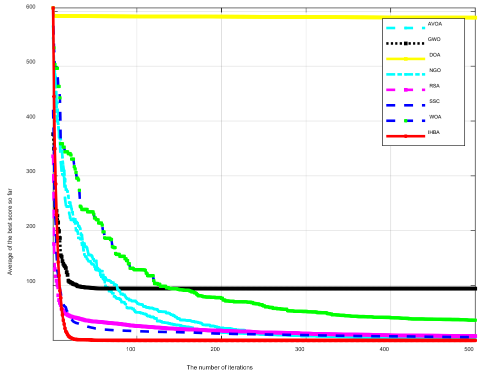
(a) CEC1



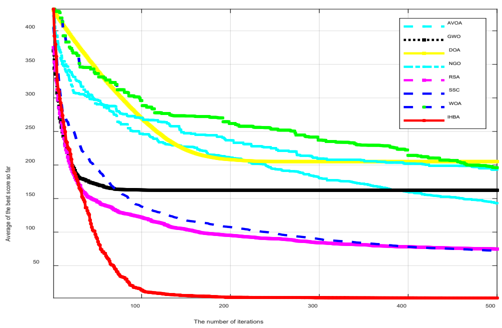
(b) CEC5



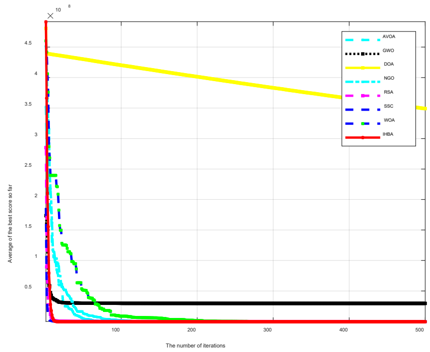
(c) CEC10



(d) CEC12



(e) CEC19



(f) CEC20

Figure 2. Convergence curves of different algorithms on the selected functions e.g., (a) CEC1, (b) CEC5, (c) CEC10, (d) CEC12, (e) CEC19, and (f) CEC20 with D is set to 50

Figure 2 shows the convergence curves of the proposed IHBA algorithm in comparison with the AVOA, GWO, NGO, RSA, SSC, and WOA algorithms for several selected test functions. It can be seen that the proposed IHBA algorithm has a faster convergence speed than the other algorithms in comparison. In general, under the classical mathematical test functions, the performance of the IHBA algorithm is better than several algorithms shown in the comparison.

4. IHBA for Coverage Optimization in WSN

This section presents the implementation of the optimal nodes coverage of the WSN deployment based on the IHBA algorithm. The subsection of the majority processing steps and analyses & discuss results are stated as follows.

4.1 Coverage Optimization Strategy

The ideal solution to the coverage optimization problem is the goal location of each node deployed. The location-seeking process of nodes is abstracted as the process of making varied movement behaviors of the honey badger group toward food or a specific site. The purpose of WSN coverage optimization utilizing the IHBA approach is to optimize the coverage of the target monitoring area by using a limited number of sensor nodes and optimizing their deployment locations. The objective function is calculated by maximizing the coverage ratio, which is max the ratio of probability to the network deployed surface 2D WSN monitoring area. The according formula of Eq. (4) is maximized as given following.

$$Maximize\,Cov_R = \frac{\sum_{j=1}^{N} P\left(S,T_j\right)}{W \cdot L}, \tag{17}$$

where Cov_R and $P(S,T_j)$ are the coverage ratio of WSN nodes, and the probability of the target point reaching $W{\cdot}L$ sensed node monitoring 2D network deployed area. Each badger individual in the algorithm represents a coverage distribution, which means the coverage distribution is mapped to solution X of the IHBA optimization. So, the objective function of WSN nodes coverage optimization is expressed in Eq.(17) by mapping it with F(X) maximizing. The specific algorithm steps are of the algorithm scheme for the coverage optimization are listed as follows.

Step 1. Input parameters such as a number of nodes M, perception radius R_s, area of region $W{\cdot}L$, etc.

Step 2. Set the parameters of population size N, the maximum number of iterations *max_Iter*, density factor, and prey attraction, randomly initialize the honey badger positions as Eqs. (5), (6).

Step 3. Enhance the initializing population by using the elite reverse learning strategy Eq.(14), and calculate the objective function for initial coverage according to Eq. (17).

Step 4. Update the motion direction Eq.(15), the position of badgers the strategy according to the Eq.(16), Then compare them to select the best fitness value according to the objective function value.

Step 5. Calculate the individual value of honey badgers and retain the optimal solution of the global best

Step 6. Determine whether the end condition is reached, if yes, proceed to the next step, otherwise go to Step 4.

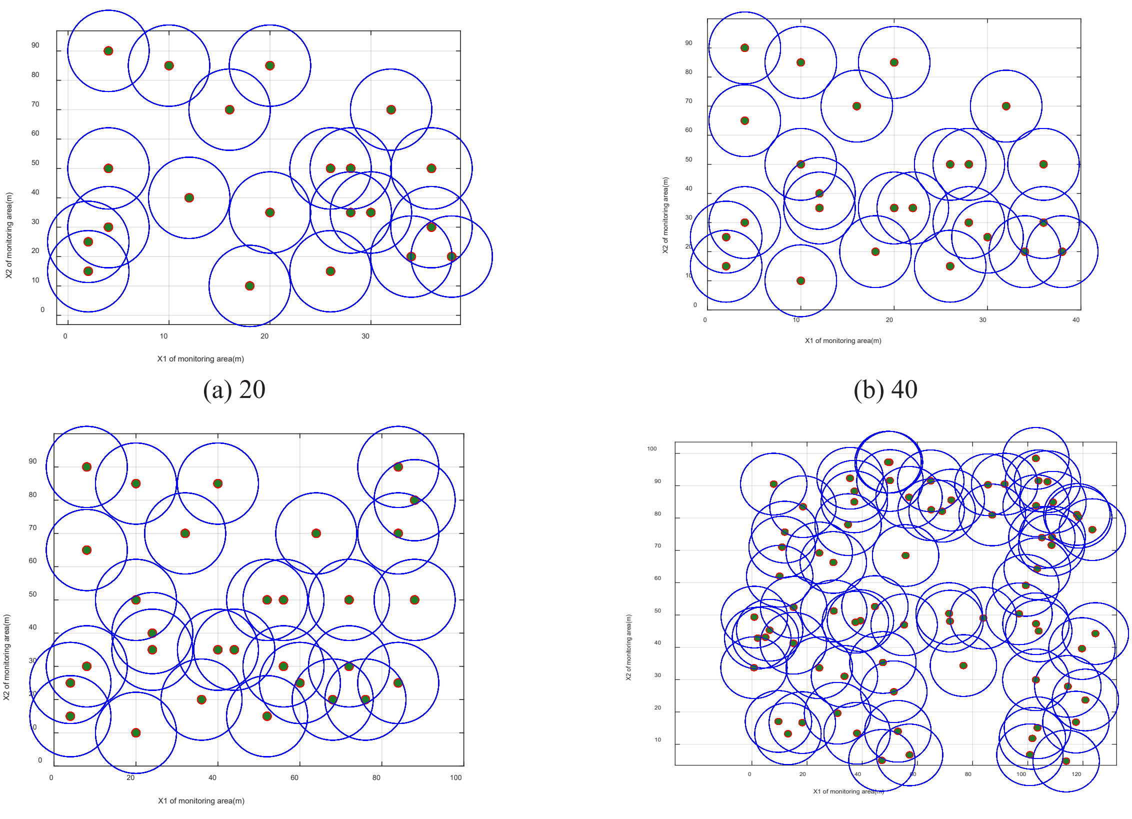
Step 7. The program ends and outputs the optimal fitness value and the honeybadger best location; it means the node's optimal coverage rate outputs.

4.2 Analysis and Discussion Results

The scenarios of assuming that WSN's sensor nodes are deployed in a square monitoring area of $W{\cdot}L$ can be set to scenario areas, e.g., 40*m*×40*m*, 80*m*×80*m*, 100*m*×100*m* and 160*m*×160*m*. Table 5 lists the experimental parameters of the WSN node deployment areas; sensing radius of sensor nodes R_S is set to 10 m; communication radius R_C is set to 20 *m*; the number of sensor nodes denotes M, consisting of 20, 40, 50, and 60 sensor nodes, respectively. *Iter* indicates the number of iterations that may be set to 500, 1000, and 1500, respectively.

Table 5. The parameter settings for the WSN node deployment region

Description	Parameters	Values
Area of deployment	$W{\cdot}L$	40m×40m, 80m×80m, 100m×100m, 160m×160m
Sensing radius	R_s	10 m
Communication radius	R_c	20 m
Number of sensor nodes	M	20, 40, 50, 60
Number of iterations	*Iter*	500, 1000,1500



(a) 20 (b) 40

(c) 50 (d) 60

Figure 3. The initialization graphical coverage diagram of the IHBA for the statistical coverage optimization scheme with different m sets of the number of sensor nodes to (a) 20, (b) 40, (b) 50, and (c) 60, respectively

The optimal results of the IHBA are compared with the selected other schemes, e.g., the SSA [52], PSO [53], GWO [46], SCA [54], and HBA [18], for the coverage optimization of WSN node deployment to verify the adequate performance of the algorithm. Figure 3 shows the initialization graphical coverage diagram of the IHBA for the statistical coverage optimization scheme with M is set to 20, 40, 50, and 60, respectively.

Table 6 compares the proposed IHBA approach to other strategies, e.g., the SSA, PSO, GWO, SCA, and HBA algorithms, in terms of percentage coverage rate, running times, convergence iterations, and monitoring region sizes. It is seen that the IHBA scheme produces the best global solution in the coverage areas, with a high coverage rate, coverage of the node's whole space area, and a faster time consumption than the other approaches.

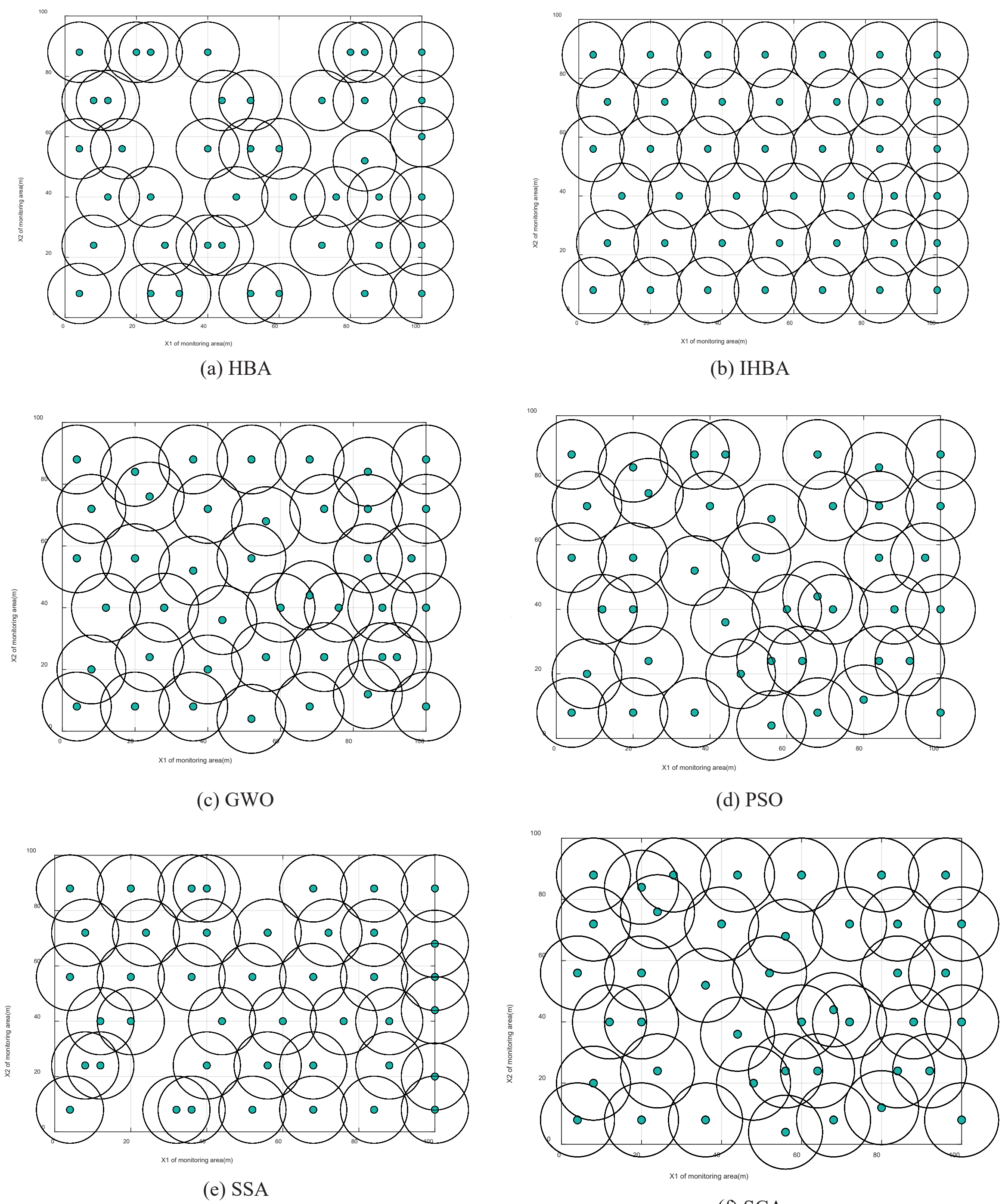
Table 6. Comparison the proposed IHBA method obtained with the other techniques: the SAA, PSO, GWO, SCA, and HBA algorithms, in different situations such as percentage coverage rate, running times, iterations to convergence and monitoring region sizes

Approach	Factor variables	40m×40m	80m×80m	100m×100m	160m×160m
SSA	Coverage rate (%)	78%	74%	77%	74%
	Time consumption (s)	3.19E+00	6.94E+00	8.28E+00	9.14E+00
	No. of iterations to convergence	145	256	234	844
	No. of sensor nodes	20	40	50	60
PSO	Coverage rate (%)	79%	77%	79%	76%
	Time consumption (s)	2.98E+00	6.32E+00	8.15E+00	8.41E+00
	No. of iterations to convergence	396	343	343	754
	No. of sensor nodes	20	40	50	60
GWO	Coverage rate (%)	80%	80%	84%	78%
	Time consumption (s)	3.26E+00	6.84E+00	8.01E+00	9.25E+00
	No. of iterations to convergence	334	44	544	755
	No. of sensor nodes	20	40	50	60
CSA	Coverage rate (%)	79%	79%	83%	78%
	Time consumption (s)	2.98E+00	6.28E+00	8.22E+00	9.22E+00
	No. of iterations to convergence	445	555	665	876
	No. of mobile nodes	20	40	50	60
HBA	Coverage rate (%)	80%	79%	80%	79%
	Time consumption (s)	2.92E+00	6.98E+00	8.40E+00	9.44E+00
	No. of iterations to convergence	665	333	563	954
	No. of sensor nodes	20	40	50	60
IHBA	Coverage rate (%)	**80%**	**82%**	**87%**	**80%**
	Time consumption (s)	**2.75E+00**	**6.45E+00**	**7.87E+00**	**9.19E+00**
	No. of iterations to convergence	**135**	**503**	**556**	**765**
	No. of sensor nodes	20	40	50	60

Figure 4 indicates graphical coverage of six different metaheuristic algorithms, e.g., the HBA, SSA, PSO, GWO, SCA, and IHBA approaches for the WSN node areas deployment scenarios for optimal coverage rates with the same density and condition environment setting. Because the IHBA algorithm can avoid premature phenomena, the coverage rate is reasonably high, with less overlap. It can better alter the node configuration than the other competitors for the monitoring region's network coverage.

Figure 5 indicates four different sizes WSN monitoring node regions deployment scenarios of the metahueric approaches for optimal coverage rates. The convergence curves of the proposed IHBA approach can provide higher percentages statical coverage than the other methods in the cases.

Figure 6 displays the coverage rate of the IHBA optimization compared against the SSA, PSO, GWO, SCA, and HBA algorithms for statistical sensor node counts deployment on the 2D monitoring different areas cases. It can be seen the IHBA algorithm produces a coverage rate that is reasonably high in the monitoring area's network coverage. The results show the IHBA approach provides the coverage rate is reasonably high, with less overlap and better altered the sensor nodes configuration in average coverage rate under the same test conditions.

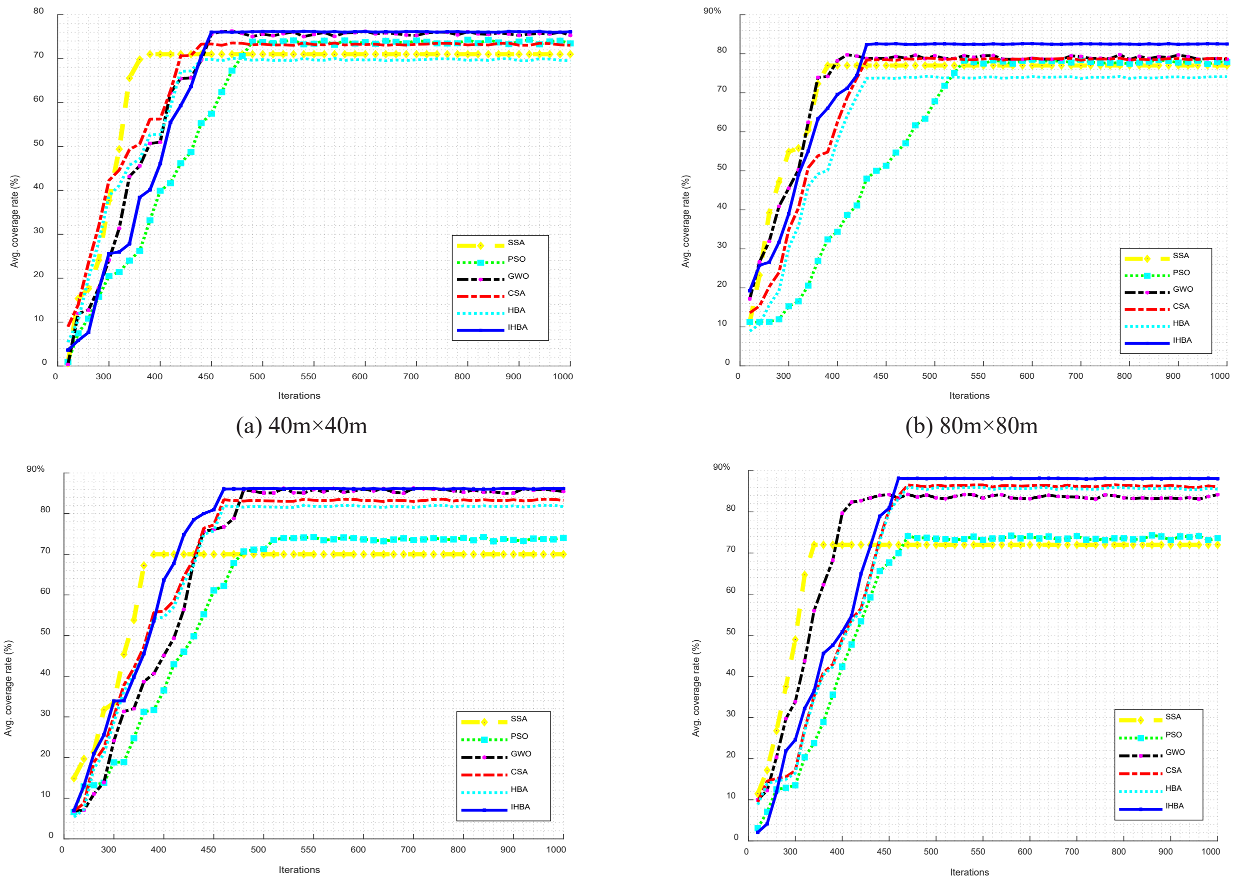


(a) HBA (b) IHBA

(c) GWO (d) PSO

(e) SSA (f) SCA

Figure 4. The graphical coverage of six different metaheuristic algorithms: the HBA, SSA, PSO, GWO, SCA, and IHBA approaches for the WSN node areas deployment



(a) 40m×40m

(b) 80m×80m

(c) 100m×100m

(d) 160m×160m

Figure 5. Four different sizes WSN monitoring node areas deployment scenarios of the metahueric approaches for optimal coverage rates

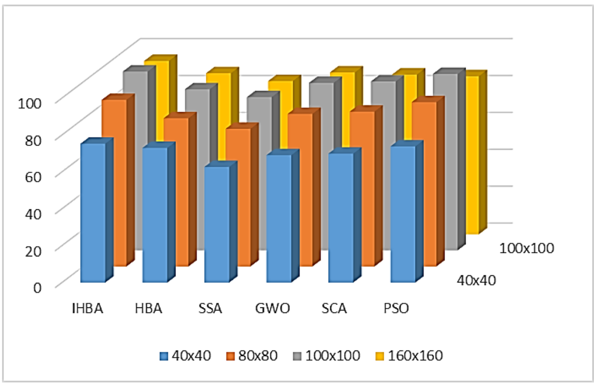


Figure 6. Comparison of the coverage rate of the IHBA optimization for sensor node coverage deployment on the 2D monitoring of the various areas cases

5. Conclusion

This study proposed an improved Honey-badger algorithm (IHBA) to address the problems of the wireless sensor network (WSN) nodes' uneven distribution and low coverage in the random deployment. The implement IHBA was conducted by modifying updating equations with elite reverse learning and multi-direction strategies to avoid the drawbacks, e.g., slow convergence speed, ease to fall into local extremum whenever dealing with the complicated situations of the original honey badger algorithm (HBA). The objective function of the optimal node coverage is modeled mathematically by calculating the distance between nodes by measuring the sensing radius of each sensor node and its capability of communication in deploying WSN. Compared results of optimal findings on the selected benchmark functions and the WSN node coverage show that the proposed IHBA makes effective the optimal solution to both coverage and benchmark problems. Significantly, the coverage rate archived of the IHBA is 87%, but the other methods are only below or equally 84% in the same comparison conditions. The suggested IHBA will apply to optimal WSN localization or deployment in future work.

Acknowledgments

This study was partially supported by the VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, *Computer Networks*, Vol. 52, No. 12, pp. 2292-2330, August, 2008.

[2] M. -F. Othman, K. Shazali, Wireless sensor network applications: A study in environment monitoring system, *Procedia Engineering*, Vol. 41, pp. 1204-1210, 2012.

[3] T.-T. Nguyen, T.-K. Dao, M.-F. Horng, C.-S. Shieh, An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks, *Journal of Network Intelligence*, Vol. 1, No. 1, pp. 23-37, February, 2016.

[4] T.-T. Nguyen, T.-K. Dao, H.-Y. Kao, M.-F. Horng, C.-S. Shieh, Hybrid Particle Swarm Optimization with Artificial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *Journal of Internet Technology*, Vol. 18, No. 4, pp. 743-752, July, 2017.

[5] R. Mulligan, H. M. Ammari, Coverage in Wireless Sensor Networks: A Survey, *Network Protocols and Algorithms*, Vol. 2, No. 2, pp. 27-53, June, 2010.

[6] G. Fan, S. Jin, Coverage Problem in Wireless Sensor Network: A Survey, *Journal of Networks*, Vol. 5, No. 9, pp. 1033-1040, September, 2010.

[7] D. K. Chaudhary, R. L. Dua, Application of multi objective particle swarm optimization to maximize coverage and lifetime of wireless sensor network, *International Journal of Computational Engineering Research*, Vol. 2, No. 5, pp. 1628-1633, September, 2012.

[8] T. T. Nguyen, J. S. Pan, T. K. Dao, An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network, *IEEE Access*, Vol. 7, pp. 75985-75998, June, 2019.

[9] Y. Qiao, T. K. Dao, J. S. Pan, S. C. Chu, T. T. Nguyen, Diversity teams in soccer league competition algorithm for wireless sensor network deployment problem, *Symmetry*, Vol. 12, No. 3, Article No. 445, March, 2020.

[10] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, *Information Sciences*, Vol. 237, pp. 82-117, July, 2013.

[11] N. Ghorbani, E. Babaei, Exchange market algorithm, *Applied Soft Computing*, Vol. 19, pp. 177-187, June, 2014.

[12] Q. Askari, I. Younas, M. Saeed, Political Optimizer: A novel socio-inspired meta-heuristic for global optimization, *Knowledge-Based Systems*, Vol. 195, Article No. 105709, May, 2020.

[13] X.-S. Yang, Harmony search as a metaheuristic algorithm, in: Z. W. Geem (Eds.), *Music-inspired harmony search algorithm*, Springer, Verlag Berlin Heidelberg, 2009, pp. 1-14.

[14] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, *2007 IEEE congress on evolutionary computation*, Singapore, 2007, pp. 4661-4667.

[15] R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, *CAD Computer-Aided Design*, Vol. 43, No. 3, pp. 303-315, March, 2011.

[16] Y. Shi, R. Eberhart, A modified particle swarm optimizer, *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, Anchorage, AK, USA, 1998, pp. 69-73.

[17] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[18] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, W. Al-Atabany, Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems, *Mathematics and Computers in Simulation*, Vol. 192, pp. 84-110, February, 2022.

[19] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, Vol. 114, pp. 163-191, December, 2017.

[20] H. Peraza-Vázquez, A. F. Peña-Delgado, G. Echavarría-Castillo, A. B. Morales-Cepeda, J. Velasco-Álvarez, F. Ruiz-Perez, A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies, *Mathematical Problems in Engineering*, Vol. 2021, Article No. 9107547, September, 2021.

[21] B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, *Computers & Industrial Engineering*, Vol. 158, Article No. 107408, August, 2021.

[22] M. Dehghani, Š. Hubálovský, P. Trojovský, Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems, *IEEE Access*, Vol. 9, pp. 162059-162080, December, 2021.

[23] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, A. H. Gandomi, Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer, *Expert Systems with Applications*, Vol. 191, Article No. 116158, April, 2022.

[24] G. Dhiman, SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications, *Knowledge-Based Systems*, Vol. 222, Article No. 106926, June, 2021.

[25] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, *Advances in Engineering Software*, Vol. 95, pp. 51-67, May, 2016.

[26] P. J. M. Van Laarhoven, E. H. L. Aarts, Simulated annealing, in: *Simulated annealing: Theory and applications*, Springer, 1987, pp. 7-15.

[27] S. Kumar, D. Datta, S. K. Singh, Black hole algorithm and its applications, in: A. Azar, S. Vaidyanatham (Eds.), *Computational intelligence applications in modeling and control*, Springer, Cham, 2015, pp. 147-170.

[28] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowledge-based systems*, Vol. 96, pp. 120-133, March, 2016.

[29] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, *Computers & structures*, Vol. 112-113, pp. 283-294, December, 2012.

[30] M. Srinivas, L. M. Patnaik, Genetic Algorithms: A Survey, *Computer*, Vol. 27, No. 6, pp. 17-26, June, 1994.

[31] J. R. Koza, *Genetic programming*, MIT press Cambridge, 1994.
[32] R. Storn, K. Price, Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report No. TR-95-012, March, 1995.
[33] D. Simon, Biogeography-Based Optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, pp. 702-713, December, 2008.
[34] D. B. Fogel, L. J. Fogel, J. W. Atmar, Meta-evolutionary programming, *Conference record of the twenty-fifth asilomar conference on signals, systems & computers*, Pacific Grove, CA, USA, 1991, pp. 540-545.
[35] N. Moosavian, B. K. Roodsari, Soccer League Competition Algorithm: A Novel Meta-heuristic Algorithm For Optimal Design of Water Distribution Networks, *Swarm and Evolutionary Computation*, Vol. 17, pp. 14-24, August, 2014.
[36] A. H. Kashan, League championship algorithm: a new algorithm for numerical function optimization, *2009 international conference of soft computing and pattern recognition*, Malacca, Malaysia, 2009, pp. 43-48.
[37] R. Moghdani, K. Salimifard, Volleyball premier league algorithm, *Applied Soft Computing*, Vol. 64, pp. 161-185, March, 2018.
[38] M. Dehghani, M. Mardaneh, J. M. Guerrero, O. Malik, V. Kumar, Football game based optimization: An application to solve energy commitment problem, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 514-523, October, 2020.
[39] P. R. Singh, M. Abd Elaziz, S. Xiong, Ludo game-based metaheuristics for global and engineering optimization, *Applied Soft Computing*, Vol. 84, Article No. 105723, November, 2019.
[40] F. A. Zeidabadi, M. Dehghani, Poa: Puzzle optimization algorithm, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, pp. 273-281, 2022.
[41] A. K. Idrees, K. Deschinkel, M. Salomon, R. Couturier, Distributed lifetime coverage optimization protocol in wireless sensor networks, *The journal of supercomputing*, Vol. 71, No. 12, pp. 4578-4593, December, 2015.
[42] A. K. Idrees, K. Deschinkel, M. Salomon, R. Couturier, Perimeter-based coverage optimization to improve lifetime in wireless sensor networks, *Engineering Optimization*, Vol. 48, No. 11, pp. 1951-1972, 2016.
[43] A. K. Idrees, K. Deschinkel, M. Salomon, R. Couturier, Multiround distributed lifetime coverage optimization protocol in wireless sensor networks, *The Journal of supercomputing*, Vol. 74, No. 5, pp. 1949-1972, May, 2018.
[44] B. Wang, Coverage problems in sensor networks: A survey, *ACM Computing Surveys (CSUR)*, Vol. 43, No. 4, pp. 1-53, October, 2011.
[45] A. K. Idrees, W. L. Al-Yaseen, Distributed genetic algorithm for lifetime coverage optimisation in wireless sensor networks, *International Journal of Advanced Intelligence Paradigms*, Vol. 18, No. 1, pp. 3-24, 2021.
[46] Z. Wang, H. Xie, Z. Hu, D. Li, J. Wang, W. Liang, Node coverage optimization algorithm for wireless sensor networks based on improved grey wolf optimizer, *Journal of Algorithms & Computational Technology*, Vol. 13, Article No. 1748302619889498, January-December, 2019.
[47] A. Singh, S. Sharma, J. Singh, Nature-inspired algorithms for Wireless Sensor Networks: A comprehensive survey, *Computer Science Review*, Vol. 39, Article No. 100342, February, 2021.
[48] A. Singh, S. Sharma, J. Singh, R. Kumar, Mathematical modelling for reducing the sensing of redundant information in WSNs based on biologically inspired techniques, *Journal of Intelligent & Fuzzy Systems*, Vol. 37, No. 5, pp. 6829-6839, November, 2019.
[49] X. Xia, J. Liu, Y. Li, Particle Swarm Optimization Algorithm with Reverse-Learning and Local-Learning Behavior, *Journal of software*, Vol. 9, No. 2, pp. 350-357, February, 2014.
[50] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.
[51] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-based systems*, Vol. 89, pp. 228-249, November, 2015.
[52] J. Chelliah, N. Kader, Optimization for connectivity and coverage issue in target-based wireless sensor networks using an effective multiobjective hybrid tunicate and salp swarm optimizer, *International Journal of Communication Systems*, Vol. 34, No. 3, Article No. e4679, February, 2021.
[53] N. A. B. Ab Aziz, A. W. Mohemmed, M. Y. Alias, A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram, *2009 international conference on networking, sensing and control*, Okayama, Japan, 2009, pp. 602-607.
[54] F. Fan, S.-C. Chu, J.-S. Pan, Q. Yang, H. Zhao, Parallel Sine Cosine Algorithm for the Dynamic Deployment in Wireless Sensor Networks, *Journal of Internet Technology*, Vol. 22, No. 3, pp. 499-512, May, 2021.

Biographies

Trong-The Nguyen received his Ph.D. degree in Communication Engineering from the National Kaohsiung University of Applied Sciences, Taiwan, in 2016. He is currently a researcher with MMLab, UIT, and a lecturer in the School of Computer Science and Mathematics, FJUT, China. His current research interests include computational intelligence, sensor networks, signal processing, and electrical engineering.

Thi-Kien Dao received her Ph.D. degree in Electronics and Engineering from the National Kaohsiung University of Technology and Sciences, Taiwan, in 2019. She is currently a lecturer in the School of Computer Science and Mathematics, Fujian University of Technology. Her current research interests include computational intelligence, data mining, sensor networks, and grid computing.

Trinh-Dong Nguyen received the Ph.D. degrees in Software Engineering from VNU Hanoi-University of Engineering and Technology, in 2018. He is currently a lecturer in Faculty of Software Engineering, University of Information Technology, VNUHCM, Vietnam. His current research interests include software engineering, real-time system, formal methods, computational intelligence, and machine learning.

Vinh-Tiep Nguyen received Ph. D. degrees in computer science from University of Information Technology, Vietnam Nation University-HCMC, in 2020. He is currently a lecturer at University of Information Technology, the Head of Multimedia Communications Laboratory (MMLab), VNU-HCM. His current research interests include content-based image retrieval, deep learning, and multimedia systems.