Journal of Network Intelligence ©2021 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **6**, Number **4**, November **2021**

A Dynamic Parallel Harris Hawks Optimization Based WSN Node Localization Algorithm

Xiankang He

College of Computer Science and Mathematics
Fujian University of Technology
Fuzhou 350118, China
296807842@qq.com

Lijun Yan

Shenzhen Institute of Information Technology
Shenzhen 518000, China
yanlj@sziit.edu.cn

Shu-Chuan Chu

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao 266590, China
College of Science and Engineering
Flinders University
Adelaide, Australia
scchu0803@gmail.com

Shi-Jian Liu

College of Computer Science and Mathematics
Fujian University of Technology
Fuzhou 350118, China
liusj2003@126.com

Jeng-Shyang Pan*

Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou 350118, China
College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao 266590, China
*Correponding Author:jengshyangpan@gmail.com

Received May 2021; revised August 2021

Abstract. *The localization of wireless sensor network (WSN) is an increasingly prominent problem. The goal of this problem is to use the anchor nodes in WSN to estimate the geographical location of the unknown nodes. This paper proposes a novel algorithm, named dynamic parallel Harris Hawks optimization (DPHHO). It contains the dynamic control strategy of the escaping energy and the parallel communication mechanism. DPHHO algorithm significantly improves the global search ability of the original Harris Hawks optimization (HHO) algorithm. Also, a novel localization algorithm based on hop distance correction and DPHHO is proposed. The proposed DPHHO algorithm was tested on the 23 classical test functions and the DPHHO-DV-Hop algorithm is applied to the localization of WSN. The experimental results show that, compared with HHO and other optimization algorithms, the proposed DPHHO algorithm is more effective and efficient. Compared with DV-Hop and other localization algorithms, the proposed DPHHO-DV-Hop is an effective algorithm for the localization of WSN.*
Keywords: WSN, DV-Hop, parallel, Harris Hawkes optimization

1. **Introduction.** Wireless Sensor Network (WSN) is a self-organizing network formed by multiple nodes with data collection, processing, transmission capabilities [1−4]. It involves several highly interdisciplinary. WSN can be widely used in many harsh environments. In practical applications, sensor nodes transmit the collected physical information to the control center [5−7]. For the most part, the data obtained is meaningful only when combined with location information [8]. Therefore, localization of WSN nodes has attracted more and more attention. Approaches for the localization of WSN nodes can be divided into: range-based algorithms and range-free algorithms. The angle or distance information between nodes should be necessary in the range-based approaches [9]. Angle-of-Arrival (AOA) [10], Received Signal Strength Indicator (RSSI) [11,12], Time Difference of Arrival (TDOA) [13], and Time of Arrival (TOA) [14] are several popular range-based location approaches. The range-free location technology has relatively low requirements for node hardware. Therefore, the range-free location technology is more feasible for the energy-limited WSN. The classical range-free location methods include distance vector hop (DV-Hop) [15,16], Approximate Point in Triangle Test (APIT) [17], multidimensional scaling-map (MDS-MAP) [18]. Among them, the DV-Hop algorithm is simple with low requirement of anchor node density and low communication. DV-Hop has become one of the most widely used localization algorithms [19]. The main process of the DV-Hop is composed of three steps roughly. In the first and second steps, the distances between the anchor node and each unknown node are calculated using the connectivity relationship between nodes. The third step is using the least square method to estimate the position of the unknown nodes. At present, most researchers have two main ways of improving this algorithm. One way is to improve the estimation of hop-count and average hop distance in the first and second steps. Another way is to invest more effective location algorithms in the third step. Namin et al. [20] selected particle swarm intelligence algorithm to replace the least square method. This method has the same variation characteristics as traditional DV-Hop. It can build up the location accuracy. Rajakumar et al. proposed the grey wolf optimization algorithm to solve the problem of multimodal localization [21]. This algorithm performs well in unknown node location recognition and positioning accuracy. The Parallel Whale Optimization Algorithm (PWOA) was proposed by Chai et al. [22] It includes two strategies for information exchange between populations. The strategy of information exchange between populations significantly enhances the population diversity and global search capability of the Proto-Whale Optimization Algorithm (WOA) [23]. The PWOA algorithm is used to optimize the location of the wireless sensor network. This work proposed a novel localization algorithm based on the

analysis of DV-Hop. The hop-count threshold is introduced to determine the number of anchor nodes participating in the calculation of average hop distance in the second steps of the DV-Hop, and the average hop distance is corrected by weight. Finally, the dynamic parallel Harris Hawks optimization (DPHHO) algorithm is introduced to optimize the coordinates of the nodes to be measured.

The structure of this paper is as follows. In the second section, the Harris Hawks optimization (HHO) algorithm and the DV-Hop method for WSN location are introduced. The third section describes in detail the communication strategy of DPHHO and the proposed the DPHHO-DV-Hop algorithm. In Section 4, some experiments are implemented to evaluate the performance of DPHHO and the DPHHO-DV-Hop method. A conclusion is given in Section 5.

2. **Related work.**

2.1. **Harris hawks optimization (HHO).**

Algorithms of Swarm Intelligence have been shown to be useful for optimization problem [24−26]. Heidari et al. proposes the HHO algorithm based on the prey behavior of Harris hawks [27]. The main structure of HHO algorithm is composed of the following three parts.

2.1.1. *Exploration phase.*

During the hunting process, Harris hawks perches in a random place and uses two strategies to find prey.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{1}$$

where $q \in [0, 1]$ as a random number that converts the two strategies, t is the number of iterations, $X(t)$ and $X(t+1)$ denote where individual will appear in the current iteration and the next iteration respectively, $X_{rand}(t)$ represents the position of individual randomly selected from the current population, $X_{rabbit}(t)$ denotes the prey location, that is, the individual location with the optimal fitness, r_1, r_2, r_3 and r_4 are updated in each iteration, which are random numbers belong to [0,1], LB and UB are the range of the hawk's initial random position, X_m is the current population's average position.

$$X_m(t) = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} X_i(t) \tag{2}$$

where N is the size of the population and $X_i(t)$ represents where the i-th hawk appear in the t-th iteration.

2.1.2. *Transition factor.*

The transition factor E simulates the prey's escaping energy. The change of E control the transition of the HHO algorithm in the exploration to the exploitation phase. During the escape, the prey's energy is greatly reduced. The prey's energy is positioned as:

$$E = 2E_0 \left(1 - \tfrac{t}{T}\right) \tag{3}$$

where E_0 represents the initial state of E, and T denotes the maximum number of iterations.

2.1.3. *Exploitation phase.*

Defines $r \in [0, 1]$ as a random number, which is used to select different exploitation strategies. Considering $r \geq 0.5$ and $|E| \geq 0.5$, adopt soft besiege strategy to update individual location:

$$X(t+1) = \triangle X(t) - E \left| J X_{rabbit}(t) - X(t) \right| \tag{4}$$

$$\triangle X(t) = X_{rabbit}(t) - X(t) \tag{5}$$

where $\triangle X(t)$ represents distance between the current position of hawk and the rabbit in the t-th iteration, $J = 2(1 - r_5)$ denotes the random moving step of the prey, $r_5 \in [0, 1]$ is a random number. Considering $r \geq 0.5$ and $|E| < 0.5$, adopt hard besiege strategy to update individual location:

$$X(t+1) = X_{rabbit}(t) - E\,|\triangle X(t)| \tag{6}$$

Considering $r < 0.5$ and $|E| \geq 0.5$, adopt soft besiege with progressive rapid dives strategy to update individual location:

$$X(t+1) = \begin{cases} Y & \text{if } f_{it}(Y) < f_{it}(X_t) \\ Z & \text{if } f_{it}(Z) < f_{it}(X_t) \end{cases} \tag{7}$$

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X(t)| \tag{8}$$

$$Z = Y + S \times LF(D) \tag{9}$$

where D denotes the dimension of solving the problem, f_{it} is the fitness function, LF denotes the Levy distribution function, S represents a random vector by size $1 \times D$. Considering $r < 0.5$ and $|E| < 0.5$, adopt hard besiege with progressive rapid dives strategy to update individual location:

$$X(t+1) = \begin{cases} Y & \text{if } f_{it}(Y) < f_{it}(X_t) \\ Z & \text{if } f_{it}(Z) < f_{it}(X_t) \end{cases} \tag{10}$$

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X_m(t)| \tag{11}$$

$$Z = Y + S \times LF(D) \tag{12}$$

2.2. **DV-Hop algorithm.** Niculescu et al. proposes the DV-Hop algorithm [28]. The main localization steps are as follows.

Through flooding technology, the anchor nodes in the monitoring area broadcast a packet to the network. The structure of the packet is $\{id, x_i, y_i, hop_i\}$, including its identifier id, location coordinate (x_i, y_i), and a zero initialized hop-count value hop_i. For all the received packets that each node creates its own hop-count table. The table record the ids, localization coordinates, and hop-counts of all anchor nodes. id is used to determine if the packet has been received in the table before, and if not, the anchor node will be recorded in the node's table. If the packet has been received and the value of record in the table is greater than the value of hop-count in the packet, the table is updated. After that, the value of minimum hop-count is retained, and the hop value will be increased by 1 to form a new data packet. The new packet forwarded to the network again. Otherwise, then the packet is discarded. All nodes in the entire network will obtain the corresponding the value of minimum hop-count from each anchor node by this means.

According to the messages of hop-count recorded, calculate the average hop distance of itself by using Eq. (13)

$$Hopsize_i = \frac{\sum_{j\neq i}^{n-1} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j\neq i}^{n=1} h_{ij}} \tag{13}$$

where n represents the number of anchor nodes, the coordinates of anchor nodes i and j are (x_i, y_i) and (x_j, y_j) respectively, h_{ij} represents the value of minimum hop-count between anchor node i and j, $Hopsize_i$ is the average hop distance of anchor node i , and anchor node i will broadcast its own average hop distance information to the network. After the broadcast, each unknown node takes the average hop distance information from the nearest anchor node, and then estimates the distance between the unknown node u and each anchor node based on the hop number information obtained previously. The estimates method is given as Eq. (14):

$$d_{ui} = Hopsize_u \times h_{ui} \tag{14}$$

where d_{ui} represents the distance from the unknown node u to anchor node i, $Hopsize_u$ denotes the average hop distance selected by the unknown node, and h_{ui} represents the number of hops between the unknown node and anchor node.

After the distance between the unknown node and each anchor node is calculated, the least square method is used to solve the estimated coordinates of the unknown node.

3. **Dynamic parallel Harris Hawks optimization algorithm and its application in WSN node location.**

3.1. **Dynamic parallel Harris Hawk optimization algorithm (DPHHO).**

3.1.1. *Dynamic control strategy of the escaping energy.* In the HHO algorithm, the rabbit's escape energy determines the transition from the exploration phase to the exploitation phase. The escape energy in the original algorithm is described by parameter E. In the iterative procedure, $|E|$ decreases from 2 to 0 with the increase of iterations. When $|E|$ is greater than 1, the original algorithm performs the exploration phase; On the contrary, the algorithm uses exploitation strategy to perform the local search. Although this transition design achieves the balance between the exploration phase and the exploitation phase, $|E|$ in the second half of the iteration cannot be greater than 1 during the whole iteration process of the algorithm, leading to premature convergence and local optimal value [29]. In this paper, the escaped energy E is modified and a disturbance term is added to Equation (3). The formula is as follows:

$$\rho = randn * \left(sin^{\beta}\left(\frac{\pi}{2} * \frac{t}{T_{max}}\right) + cos\left(\frac{\pi}{2} * \frac{t}{T_{max}}\right) - 1 \right) \tag{15}$$

$$E_{new} = 2E_0 * \left(1 - \frac{t}{T_{max}} + \rho\right) \tag{16}$$

where sin and cos denote the sine and cosine functions respectively, t and T_{max} represent the current iteration and the maximum number of iterations respectively, $randn$ is a random number that obeys Gaussian distribution, β determines where the disturbance peak appears and it is a constant. According to the experimental statistics of different values of β, when β=2.5, the perturbation peak of the strategy which we proposed usually appears during the ideal target region.

3.1.2. *Parallel communication strategy.* The HHO algorithm needs to adjust fewer parameters. Improving the convergence speed and accuracy is always the driving force and goal of the optimization algorithm. The parallel communication strategy can build up the optimization precision and convergence speed of the algorithm. This strategy helps the algorithm avoid placing local states and converge to a better solution. Algorithms for parallel communication strategies generally divide populations into groups and then run the original HHO algorithm respectively. After reaching a certain number of iterations, communication is carried out, and the poor solution in each group is replaced by the better solution of other groups.

In this paper, three parallel strategies are used to communicate between groups [30] including the random solution exchange, the optimal solution exchange, and the perturbation strategy. During the update process, the algorithm randomly selects the first and second strategies. The details of the parallel communication strategy are shown in

Figure 1. Before communication, suppose that b_i is the best solution of the i-th group (i=1,...,N). Let B= [b_1,b_2,...,b_N]. The random solution exchange strategy: Choose a best solution b_j randomly from B for i-th group (i=1,...,N). If $b_i < b_j$, b_i in the i-th group will be replaced by b_j. Otherwise b_i is perturbed according to the perturbation strategy. If the solution of perturbed is better than b_i, then assign the solution of disturbed to b_i. The optimal solution exchange strategy: Suppose b_g is the global optimal solution. Perturb b_g according to the third strategy to search a better solution than b_g. If the solution of perturbed is better than b_g, then assign the solution of disturbed to b_g. The perturbation strategy is mainly realized by adding a perturbation term. Its purpose is to allow the algorithm to fully explore the local area, and the perturbation term is generated randomly by the standard normal distribution function. The following is the step flow of the DPHHO algorithm:

1 Initialize N harris hawk's position, the range of the harris hawk search space, randomly divide evenly them into G groups, and the communication step size between groups is R.
2 Each subgroup calculates the fitness function based on the position of harris hawks and chooses the position of harris hawk with the best fitness as the position of prey in the meantime.
3 Each subgroup is iterated R times according to the improved HHO algorithm and then implement the parallel communication strategy.
4 Determine whether the termination conditions are met. If not, repeat steps 2-4, otherwise terminate.

Algorithm 1 shows the pseudo-code of DPHHO.

3.2. **Hop distance correction.** When solving the distance between the node to be located and the anchor node in the traditional DV-Hop algorithm, the average hop distance of the anchor node nearest to the unknown node is used as its average hop distance. This method neglects the difference of network distribution around the anchor nodes, which will certainly bring large-ranging errors. In the DPHHO-DV-Hop, the local information of network topology is used to calculate the average hop distance [31]. For an unknown node, an anchor will be used to calculate the average hop distance if and only if the number of hops between the unknown node and the anchor node is not greater than a given threshold T. The threshold setting method is as follows:

$$\frac{1}{R}\sqrt{\frac{S\times lh}{an\times lp\times\pi}} < T < H_{max} \tag{17}$$

where S represents the area of the network, lh is the number of all anchor nodes in the network, an represents the total number of nodes in the network, and lp is the proportion of anchor nodes in the total number of nodes. R represents the radius of the transmitted power of the node. Formula (17) is the value range of T formulated under ideal circumstances, but it is difficult to achieve very uniform distribution in actual node distribution, so the value of T should be properly enhanced to meet the overall coverage of the network. For each unknown node $U_i\,(i = 1, 2, \ldots, n)$, there are $A_j(j = 1, 2, \ldots, m)$, where A_j is the anchor node within T hops. H_j is the number of hops between the unknown node and anchor node j. Suppose that the unknown node receives information from the anchor node, the weighted value of the average hop-count of each anchor node is expressed as γ_j, calculated based on Equation (18). The weighted value of A_j is the value of the reciprocal of the unknown node to A_j divided by the sum of the reciprocal hop-count of the unknown

Algorithm 1 Pseudo-code of DPHHO algorithm

Initialize the population size N and the random population $X_i (i = 1,2,\dots,N)$. Randomly divide evenly them into G groups (G is equal to 4), set the dimensions of the optimization problem dim. Current generation T=1 and the maximal number of generation F, R is the generation to trigger the communication strategy.

```
while T < F do
  for g-th =1 to G do
     Calculate the fitness values of hawks and set X_rabbit as the location of rabbit (best location)
     in the G (g-th)
        for i =1 to N/G do
           Update the energy of the rabbit E using Eq.(16);
              if (|E| > 1) then
                 Update the position using Equation (1)
              else if (|E| < 1) then
                    if (r ≥ 0.5 and |E| ≥ 0.5) then
                        Update the position using Eq.(4);
                    else if (r ≥ 0.5 and |E| < 0.5) then
                        Update the position using Eq.(6);
                    else if (r < 0.5 and |E| ≥ 0.5) then
                        Update the position using Eq.(7);
                    else if (r < 0.5 and |E| < 0.5) then
                        Update the position using Eq.(10);
                    end if
              end if
        end for
  end for
  if T = R
     Implement the parallel communication strategy
  end if
  T=T+1
end while
Output:
  The position of rabbit and its fitness value
```

node to each anchor node.

$$\gamma_j = \frac{1/H_j}{\sum_{j=1}^{m} \frac{1}{H_j}} \tag{18}$$

Let D_j be the average hop distance of A_j. The weighted average hop distance of the unknown node is calculated as Formula (19).

$$D = \sum_{j=1}^{m} \gamma_j D_j \tag{19}$$

Therefore, the weighted method can make the average distance of each hop in the network closer to the true value and achieve the effect of reducing the positioning error.

3.3. **Using DPHHO algorithm to locate the unknown nodes.** For an unknown node, after obtaining the estimated distances based on this paper's proposed method, we use the DPHHO algorithm to estimate its location. First, the objective function of DPHHO needs to be determined. The square error of distance estimation is defined as shown below,

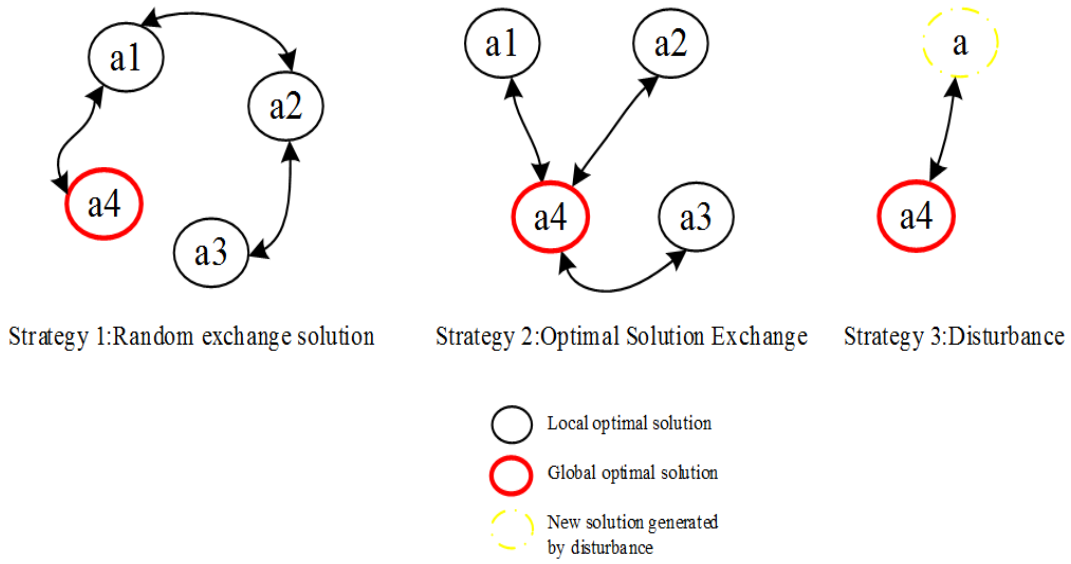


FIGURE 1. Parallel communication strategy

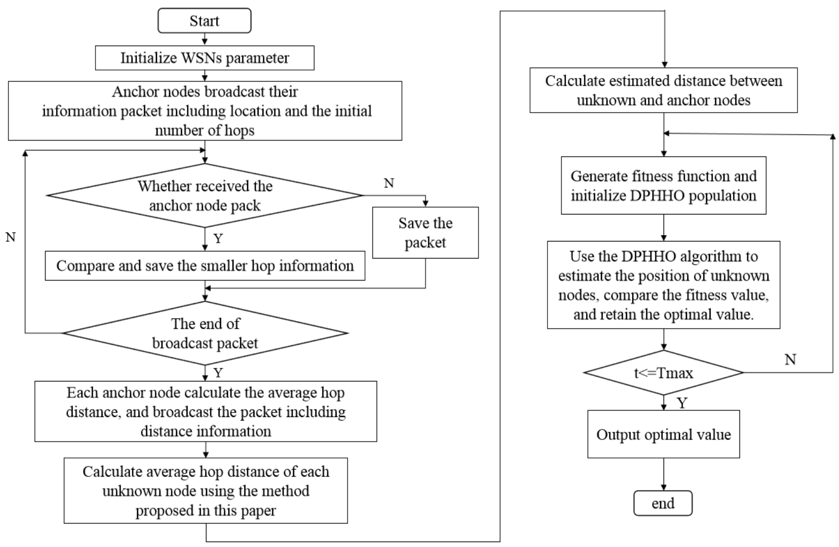


FIGURE 2. Procedure of DPHHO-DV-Hop

$$\varepsilon = \sum_{i=1}^{M_k} \left(\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} - d_{ki} \right)^2 \tag{20}$$

where M_k is the number of connected anchor nodes, (x_i, y_i) is the location of anchor node i, (x_k, y_k) is the actual location of unknown node k, and d_{ki} represents the estimated distances. Minimizing this error is the goal of the WSN localization problem. However, the estimated distance error is growth with the value of hop-count increases. The equation (21) can be weighted based on the reciprocal of the hop-count [32]. The fitness function of the DPHHO is defined as shown below by this means:

$$f(x_k, y_k) = \sum_{i=1}^{M_k} \left(\frac{1}{hop_{ki}} \right)^2 \left(\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} - d_{ki} \right)^2 \tag{21}$$

TABLE 1. Parameters of each algorithm

Algorithm	Parameter settings
PSO	pop = 40, iteration = 500, c = 2, w = [0.4,0.9]
GWO	pop = 40, iteration = 500, α = [0,2]
PMVO	pop = 40, R = 20, iteration = 500, G = 4, W_{min} = 0.2, W_{max} = 1
HHO	pop = 40, iteration = 500, E = 2
DPHHO	pop = 40, R = 20, iteration = 500, G = 4, E = 2, β = 2.5

Each unknown node uses an independent DPHHO optimizati on procedure to estimate its location. The individual encoding of DPHHO for location estimation is two-dimension variables (x_i, y_i) representing the coordinate of unknown nodes.

TABLE 2. Comparison of optimization performance of HHO, PMVO, DPHHO on 23 classical test functions

30D	**HHO**			**PMVO**			**DPHHO**		
	BEST	**MEAN**	**STD**	**BEST**	**MEAN**	**STD**	**BEST**	**MEAN**	**STD**
F1	8.44E-115	8.92E-99	2.98E-98	5.04E-01	1.13E+00	3.28E-01	**2.50E-162**	**5.66E-136**	**2.53E-135**
F2	7.65E-61	4.06E-52	1.37E-51	4.60E-01	7.48E+00	2.59E+01	**6.63E-87**	**1.67E-69**	**6.00E-69**
F3	2.35E-103	8.24E-73	3.69E-72	8.21E+01	2.08E+02	1.11E+02	**1.37E-148**	**5.29E-115**	**1.57E-114**
F4	3.01E-57	1.14E-49	3.58E-49	8.11E-01	1.46E+00	9.17E-01	**7.15E-85**	**2.97E-64**	**1.33E-63**
F5	1.59E-05	6.91E-03	9.84E-03	3.76E+01	4.49E+02	8.27E+02	**5.98E-07**	**6.07E-03**	**5.50E-03**
F6	6.02E-08	6.90E-05	1.52E-04	6.19E-01	1.17E+00	3.66E-01	**2.12E-09**	**5.87E-05**	**7.03E-05**
F7	1.35E-05	1.32E-04	**1.17E-04**	9.14E-03	2.84E-02	1.45E-02	**5.98E-06**	**1.26E-04**	1.38E-04
F8	**-1.26E+04**	**-1.26E+04**	**6.87E-01**	-9.11E+03	-7.60E+03	6.33E+02	**-1.26E+04**	**-1.26E+04**	8.09E-01
F9	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.94E+01	1.06E+02	2.25E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	8.88E-16	8.88E-16	0.00E+00	6.39E-01	1.34E+00	4.62E-01	**8.88E-16**	**8.88E-16**	**0.00E+00**
F11	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.78E-01	8.26E-01	7.12E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	5.58E-08	3.69E-06	5.44E-06	2.02E-02	9.36E-01	1.01E+00	**4.87E-09**	**3.55E-06**	**5.16E-06**
F13	1.75E-07	7.53E-05	1.21E-04	3.91E-02	1.23E-01	5.19E-02	**9.89E-08**	**5.18E-05**	**8.17E-05**
F14	**9.98E-01**	1.24E+00	1.10E+00	**9.98E-01**	1.34E+00	9.23E-01	**9.98E-01**	**1.20E+00**	**4.08E-01**
F15	**3.08E-04**	4.32E-04	**2.09E-05**	4.85E-04	**3.32E-04**	**2.09E-05**	**3.08E-04**	**3.32E-04**	2.88E-04
F16	**-1.03E+00**	**-1.03E+00**	1.60E-10	**-1.03E+00**	-1.03E+00	2.44E-07	**-1.03E+00**	**-1.03E+00**	**3.37E-11**
F17	**3.98E-01**	**3.98E-01**	4.38E-06	**3.98E-01**	3.98E-01	**8.58E-08**	**3.98E-01**	**3.98E-01**	**2.58E-06**
F18	**3.00E+00**	**3.00E+00**	2.16E-07	**3.00E+00**	**3.00E+00**	1.22E-06	**3.00E+00**	**3.00E+00**	**3.33E-08**
F19	**-3.86E+00**	**-3.86E+00**	2.61E-03	-3.86E+00	**-3.86E+00**	**5.49E-07**	**-3.86E+00**	**-3.86E+00**	1.28E-03
F20	**-3.28E+00**	-3.14E+00	7.95E-02	-3.32E+00	**-3.25E+00**	**6.11E-02**	-3.32E+00	-3.23E+00	1.02E-01
F21	-9.46E+00	-5.27E+00	**9.85E-01**	-1.02E+01	-5.31E+00	3.11E+00	**-1.01E+01**	**-7.51E+00**	1.14E+00
F22	-5.09E+00	-5.09E+00	1.43E-03	**-1.04E+01**	-5.35E+00	3.83E+00	-1.02E+01	**-6.70E+00**	**1.19E+00**
F23	-5.13E+00	-5.13E+00	**2.82E-03**	**-1.05E+01**	**-8.72E+00**	3.26E+00	**-1.05E+01**	-6.95E+00	3.74E+00
Win	13	15	13	14	16	18	—	—	—
Draw	10	8	3	7	5	0	—	—	—
Lose	0	0	7	2	2	5	—	—	—

4. **Experimental analysis.** The experimental results of the DPHHO algorithm in mathematical test function and its application in the WSN node location are given in this section.

4.1. **Experimental results on mathematic test function.** In this subsection, for purpose of verifying the performance of DPHHO, 23 classical mathematical test functions [33] are used in this experiments. PMVO (PMVO in reference [34]), PSO [35], HHO and GWO [36] were used for comparison. To ensure the fairness and accuracy of the experiment, each test function is independent run for 30 times. Table 1 shows the parameter settings for these comparison algorithms.

TABLE 3. Comparison of optimization performance of PSO, GWO, DPHHO on 23 classical test functions

30D	PSO			GWO			DPHHO		
	BEST	**MEAN**	**STD**	**BEST**	**MEAN**	**STD**	**BEST**	**MEAN**	**STD**
F1	1.13E-05	6.72E-04	9.76E-04	1.27E-32	4.77E-31	7.41E-31	**2.50E-162**	**5.66E-136**	**2.53E-135**
F2	1.04E-01	1.93E+01	4.44E+01	1.46E-19	1.43E-18	9.81E-19	**6.63E-87**	**1.67E-69**	**6.00E-69**
F3	1.99E+01	1.00E+02	1.04E+02	1.16E-09	4.25E-06	1.57E-05	**1.37E-148**	**5.29E-115**	**1.57E-114**
F4	4.61E-01	1.46E+00	6.12E-01	2.60E-08	1.26E-07	9.42E-08	**7.15E-85**	**2.97E-64**	**1.33E-63**
F5	1.73E+01	8.89E+01	6.24E+01	2.60E+01	2.69E+01	6.92E-01	**5.98E-07**	**6.07E-03**	**5.50E-03**
F6	2.43E-05	8.45E-04	1.45E-03	6.19E-05	5.40E-01	2.96E-01	**2.12E-09**	**5.87E-05**	**7.03E-05**
F7	5.86E-02	1.04E-01	3.58E-02	2.83E-04	1.41E-03	8.70E-04	**5.98E-06**	**1.26E-04**	**1.38E-04**
F8	-8.58E+03	-6.57E+03	5.78E+02	-7.53E+03	-6.40E+03	6.66E+02	**-1.26E+04**	**-1.26E+04**	**8.09E-01**
F9	3.69E+01	6.46E+01	1.83E+01	**0.00E+00**	2.02E+00	3.16E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
F10	2.44E-03	1.18E+00	7.82E-01	4.35E-14	5.70E-14	1.04E-14	**8.88E-16**	**8.88E-16**	**0.00E+00**
F11	3.41E-06	1.20E-02	1.10E-02	**0.00E+00**	4.33E-03	8.16E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	1.30E-06	6.47E-02	1.15E-01	6.72E-03	4.10E-02	1.81E-02	**4.87E-09**	**3.55E-06**	**5.16E-06**
F13	9.87E-06	2.07E-02	4.33E-02	1.00E-01	5.11E-01	2.32E-01	**9.89E-08**	**5.18E-05**	**8.17E-05**
F14	9.98E-01	3.17E+00	2.52E+00	9.98E-01	5.31E+00	4.16E+00	**9.98E-01**	**1.20E+00**	**4.08E-01**
F15	**3.07E-04**	8.90E-04	**2.33E-04**	**3.07E-04**	5.43E-03	8.85E-03	3.08E-04	**3.32E-04**	2.88E-04
F16	**-1.03E+00**	**-1.03E+00**	**2.22E-16**	**-1.03E+00**	**-1.03E+00**	1.33E-08	**-1.03E+00**	**-1.03E+00**	3.37E-11
F17	**3.98E-01**	**3.98E-01**	**0.00E+00**	**3.98E-01**	**3.98E-01**	9.75E-07	**3.98E-01**	**3.98E-01**	2.58E-06
F18	**3.00E+00**	**3.00E+00**	**1.36E-15**	**3.00E+00**	**3.00E+00**	1.86E-05	**3.00E+00**	**3.00E+00**	3.33E-08
F19	**-3.86E+00**	**-3.86E+00**	**2.26E-15**	**-3.86E+00**	**-3.86E+00**	2.48E-03	**-3.86E+00**	**-3.86E+00**	1.28E-03
F20	**-3.32E+00**	-3.27E+00	**5.98E-02**	**-3.32E+00**	-3.28E+00	6.18E-02	**-3.32E+00**	**-3.23E+00**	1.02E-01
F21	**-1.02E+01**	-4.52E+00	2.67E+00	**-1.02E+01**	**-9.39E+00**	1.86E+00	-1.01E+01	-7.51E+00	**1.14E+00**
F22	**-1.04E+01**	**-7.48E+00**	3.40E+00	**-1.04E+01**	-1.01E+01	**1.18E+00**	-1.02E+01	-6.70E+00	1.19E+00
F23	**-1.05E+01**	-5.40E+00	1.21E+00	**-1.05E+01**	**-1.05E+01**	**8.33E-04**	**-1.05E+01**	-6.95E+00	3.74E+00
Win	14	17	16	12	15	19	—	—	—
Draw	2	2	7	2	4	4	—	—	—
Lose	7	4	0	9	4	0	—	—	—

In this article, the best value, average value, and standard deviation are used to evaluate the performances of different algorithms. Table 2 tabulates the best values (BEST), mean values (MEAN), and standard deviations (STD) of each algorithm on 23 classical mathematical test functions. For each test function, the smaller the value in the table is, the better the corresponding algorithm is. At the end of the table 2 and table 3, 'Win', 'Draw', or 'Lose' give the numbers of better, same, and worse performance of the proposed DPHHO, respectively. If the performance of proposed DPHHO at corresponding item is better than that of the corresponding algorithm, the 'Win' will add one; if it's worse, the 'Lose' will add one; otherwise, the 'Draw' will add one. According to Table 2, the DPHHO algorithm has 13 better, 10 similar, 0 worse performances than the HHO algorithm from the "best" perspective, respectively. It carries 15 better, 8 similar, 0 worse performances from the "mean" perspective, respectively. It carries 13 better, 3 similar, 7 worse performances from the "standard deviation" perspective, respectively. Compared with PMVO algorithm, DPHHO algorithm has won 60.9% in the 23 classical test functions from the "best" perspective. In contrast, the DPHHO algorithm only has lost 8.7% of the PMVO algorithm in 23 test functions. From the "mean" perspective, DPHHO algorithm has won 69.7% in the 23 classical test functions. The DPHHO algorithm has lost 8.7% of the PMVO algorithm in 23 test functions. From the "standard deviation" perspective, DPHHO algorithm has won 78.3% in the 23 classical test functions. The DPHHO algorithm has lost 21.7% of the PMVO algorithm in 23 test functions. According to Table 3, the winning number of DPHHO algorithm is much higher than the winning number of PSO algorithm and GWO algorithm in 23 test functions.

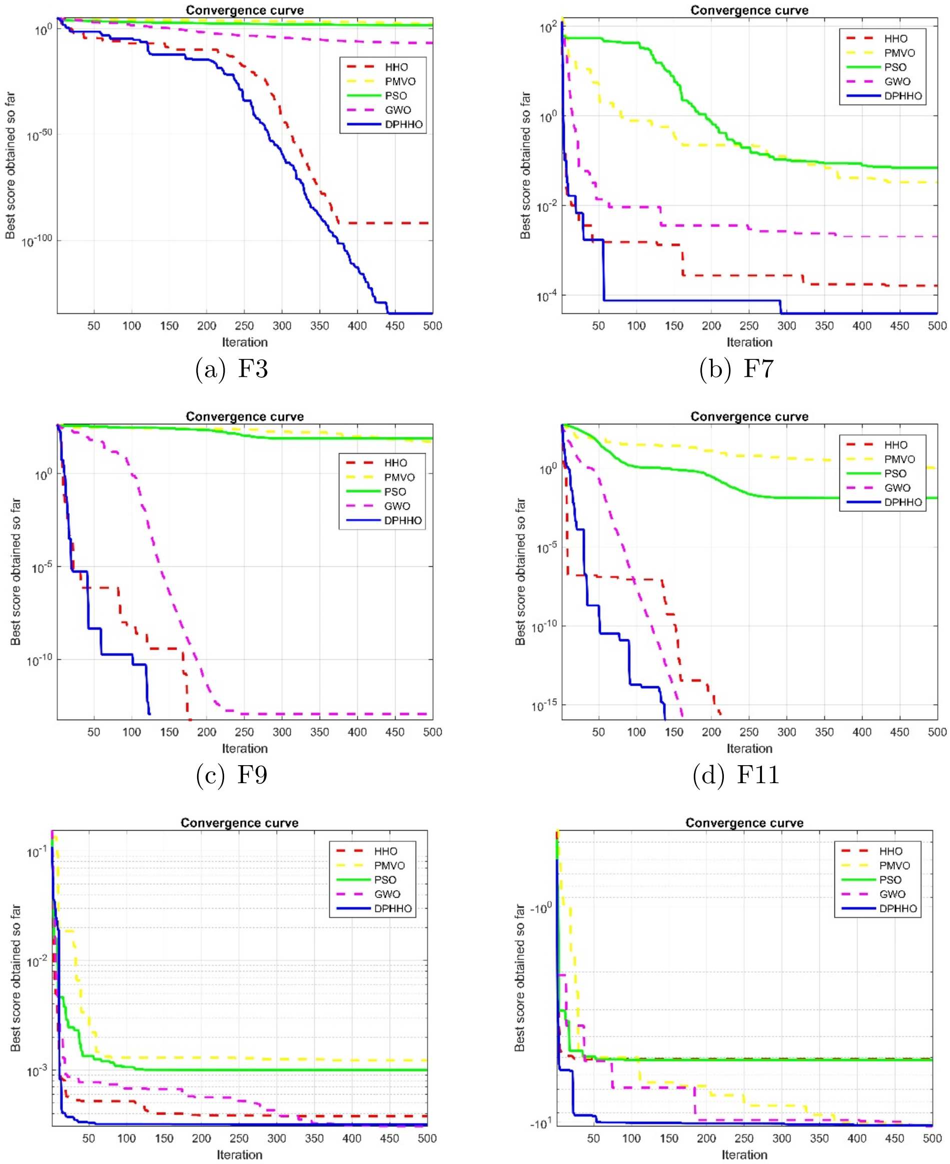
Figure 3 give the convergence curves of the several algorithms of the proposed DPHHO, HHO, PMVO, PSO, and GWO for several selected test functions. Compared with HHO, PMVO, PSO, and GWO. Our proposed DPHHO algorithm performs better convergence rates. Overall, under 23 classical mathematical test functions, the performance of the DPHHO algorithm is better than that of the compared HHO, PMVO, PMVO, and GWO algorithms.

4.2. **Experiment results of DPHHO-DV-Hop algorithm.** In order to verify the DPHHO-DV-Hop algorithm in this work, the experiment was carried out on the MATLAB 2015b platform. The hyperbolic DV-hop algorithm in [37] that used two-dimensional hyperbolic takes the place of the least square method in the traditional DV-Hop. The traditional DV-Hop , DV-Hop based on PSO, hyperbolic DV-Hop, and DPHHO-DV-Hop algorithm are used for experiment analysis, the fitness function adopts formula(21). The parameters of PSO are $c1 = c2 = 2.05$, W_{max}=0.9, and W_{min}=0.4 and the maximum number of iterations is 100 and the initial population size is 20. For purpose of reducing the impact of random errors, the average value of 20 experiments was used to value the effectiveness of the algorithm. In this experiments, nodes were casually arranged in a 100 m$*$100 m network. Assuming that the communication radius of each node is R, the localization performance of the algorithm is evaluated based on the normalized relative error formula (22):

$$TALE = \frac{\sum_{i=1}^{N_k} \sqrt{(x_i - x_{ik})^2 + (y_i - y_{ik})^2}}{N_k \times R} \tag{22}$$

where the criterion for evaluating localization effect is relative localization error $TALE$, N_k is the number of unknown nodes, (x_i, y_i) denotes the estimated coordinates of the positioning calculation, and (x_{ik}, y_{ik}) denotes the real coordinates of unknown nodes.

4.2.1. *The influence of anchor node ratio on localization performance.* When the total number of nodes is 100 and R is set to 30 m, the ratio of anchor nodes is changed for experimental. The Figure 4 makes clear the experimental results. The average localization error of DPHHO-DV-Hop algorithm is about 4.9%, 20.1% and 33.3% lower than errors of the PSO-DV-Hop, Hyperbolic-DV-Hop and traditional DV-Hop algorithm, respectively. Where the average localization error is the average error under different anchor node ratios. As the ratio of anchor enhances, the localization error of the algorithm in the figure



(a) F3 (b) F7

(c) F9 (d) F11

(e) F15 (f) F22

FIGURE 3. Convergence curves of different algorithms on F3(a), F7(b), F9(c), F11(d), F15(e), F22(f) with 30D

reduces to varying degrees. Because the ratio of anchor node increases. It can improve the accuracy of hop distance estimation and provide more reliable distance information for the third stage.

4.2.2. *The influence of communication radius on localization performance.* When the total number of nodes is 100 and the ratio of anchor nodes remains at 30%. The experimental results are shown in the Figure 5. The average localization error of our proposed algorithm is about 3.3%, 19.5% and 55.0% lower than errors of the PSO-DV-Hop, Hyperbolic-DV-Hop and traditional DV-Hop algorithm, respectively. Where the average localization error is the average error under different communication radius. When R is less than 40 m, the localization error decreases obviously with the increase of the communication radius. However, when R is greater than 40 m, the decrease in localization error tends to be flat, and there is a slight upward trend. Because excessive communication radius will increase

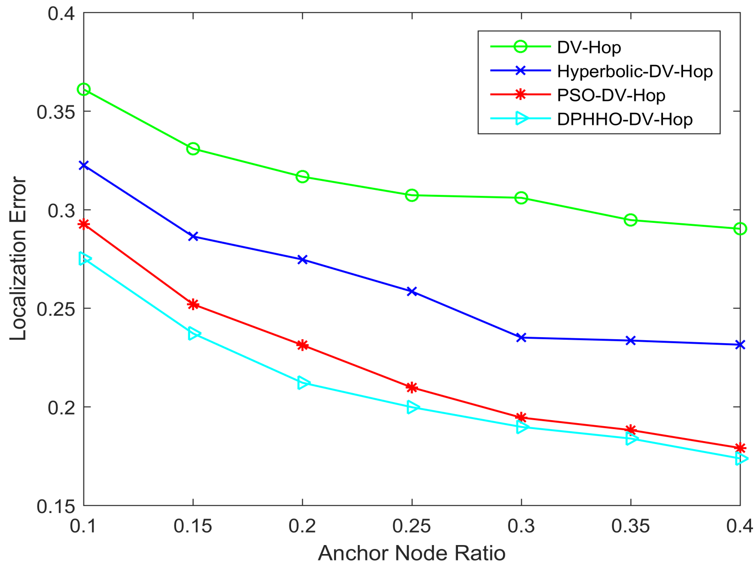


Figure 4. Relation diagram anchor node and localization error

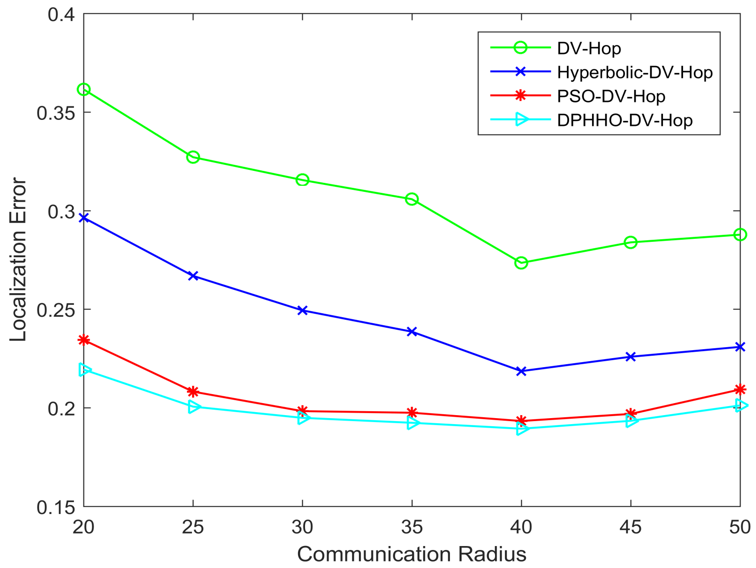


Figure 5. Relation diagram communication radius and localization error

the error of each hop distance of the anchor node, resulting in a decrease in localization accuracy.

4.2.3. *The impact of the total number of nodes on localization performance.* In this experimental that the ratio of anchor nodes remains at 30% and R is 30 m. The Figure 6 makes clear the experimental results. The average localization error of DPHHO-DV-Hop algorithm is about 2.4%, 31.5% and 45.6% lower than the PSO-DV-Hop, Hyperbolic-DV-Hop and traditional DV-Hop algorithm, respectively. Where the average localization error is the average error under different the total number of nodes. The total number of nodes increases continuously that will reduce the distance between nodes, making the estimation of average hop distance and hop number more accurate.

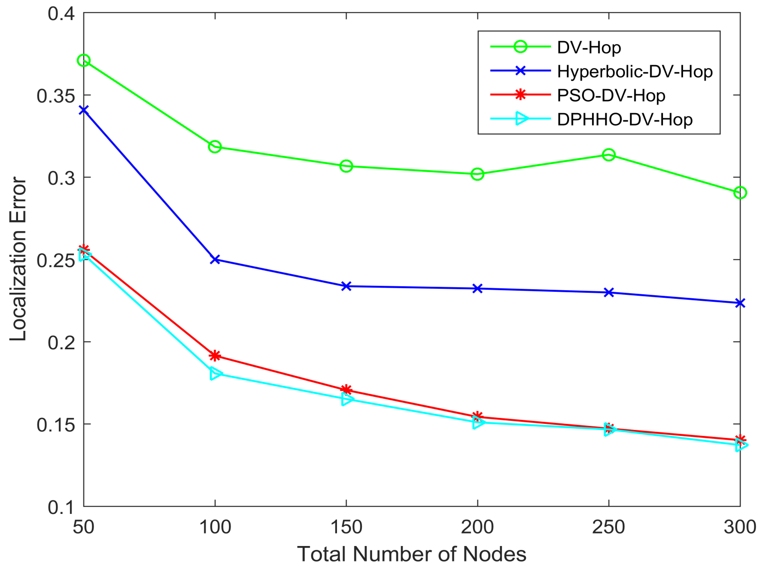


FIGURE 6. Relation diagram communication radius and localization error

5. **Conclusions.** In this work, a Dynamic Parallel Harris Hawk Optimization (DPHHO) algorithm is proposed to solve the WSN localization problem. The dynamic control strategy of the escaping energy and the parallel communication mechanism used in DPHHO are helpful to avoid placing local optimal states. The searching capability of DPHHO algorithm is improved effectively. The 23 classical test functions were used to verify the performance of the DPHHO algorithm. The experimental results of different algorithms demonstrate that the DPHHO algorithm has better performance on both converging rates and finding the better solution. A novel localization algorithm is proposed in this work. The DPHHO-DV-Hop algorithms uses the local information of the network topology when calculating the average hop distance. Several scenarios of experiments were carried out, for instance, the different ratios of anchor nodes, the diverse communication range, and the different number of the sensor node. Experimental results confirmed the performance of the propose DPHHO-DV-Hop algorithm.

Acknowledgment. This work is supported by the National Natural Science Foundations of China (No. 61872085), the Scientific Research Project of Fujian Education Department (JK2017029, JAT190069), and the Scientific Research and Development Foundation of Fujian University of Technology (GY-Z18181, GY-Z20068, XF-X19017)

REFERENCES

[1] S. R. Jino Ramson, D. Jackuline Moni, Applications of Wireless Sensor Networks – A Survey, *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, Coimbatore, India, pp. 325-329, 2017.

[2] J. S. Pan, L. P. Kong, T. W. Sung, P. W. Tsai and V. Snasel, $\alpha-$Fraction First Strategy for Hierarchical Wireless Sensor Networks, *Journal of Internet Technology*, vol. 19, no. 6, pp. 1717-1726, 2018.

[3] T. Y. Wu, L. Yang, Z. Lee, S. C. Chu, S. Kumari, S. Kumar, A Provably Secure Three-factor Authentication Protocol for Wireless Sensor Networks, *Wireless Communications and Mobile Computing*, vol. 2021, 5537018, 2021.

[4] C. M. Chen, B. Xiang, T. Y. Wu, K. H. Wang, An Anonymous Mutual Authenticated Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks, *Applied Sciences*, vol. 8, 1074, 2018.

[5] J. N. Chen, F. M. Zou, T. Y. Wu, Y. P. Zhou, A New Certificate-Based Aggregate Signature Scheme for Wireless Sensor Networks, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 5, pp. 1264-1280, 2018.

[6] L. Shkurti, X. Bajrami, E. Canhasi, B. Limani, S. Krrabaj, A. Hulaj, Development of Ambient Environmental Monitoring System Through Wireless Sensor Network (WSN) Using NodeMCU and "WSN Monitoring", *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, Bar, Montenegro, pp. 1-5, 2017.

[7] J. Zhang, H. R. Nian, X. C. Ye, X. R. Ji and Y. He, A Spatial Correlation Based Partial Coverage Scheduling Scheme in Wireless Sensor Networks, *Journal of Network Intelligence*, vol. 5, no. 2, pp. 34-43, May 2020.

[8] T. T. Nguyen, J. S. Pan, T. K. Dao, A Compact Bat Algorithm for Unequal Clustering in Wireless Sensor Networks, *Applied Sciences*, vol. 9, no. 10, 1973, 2019.

[9] 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, 2019.

[10] R. Peng, M. L. Sichitiu, Angle of Arrival Localization for Wireless Sensor Networks, *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, Reston, VA, USA, pp. 374-382, 2006.

[11] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, A Novel Approach to Indoor RSSI Localization by Automatic Calibration of the Wireless Propagation Model, *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, pp. 1-5, 2009.

[12] Z. G. Du, J. S. Pan, S. C. Chu, H. J. Luo, P. Hu, Quasi-Affine Transformation Evolutionary Algorithm With Communication Schemes for Application of RSSI in Wireless Sensor Networks,*IEEE Access*, vol. 8, pp. 8583-8594, 2020.

[13] K. C. Ho, Y. T. Chan, Solution and Performance Analysis of Geolocation by TDOA, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1311-1322, October, 1993.

[14] M. Khalaf-Allah, Time of Arrival (TOA)-Based Direct Location Method, *2015 16th International Radar Symposium (IRS)*, Dresden, Germany, pp. 812-815, 2015.

[15] N. Liu, J. S. Pan, J. Wang, T. T. Nguyen, An Adaptation Multi-Group Quasi-Affine Transformation Evolutionary Algorithm for Global Optimization and Its Application in Node Localization in Wireless Sensor Networks, *Sensors*, vol. 19, no. 19, 4112, September, 2019.

[16] C. M. Wu, T. Yang, An Improved DV-HOP Algorithm was Applied for The Farmland Wireless Sensor Network, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 148-155, January, 2017.

[17] T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. Abdelzaher, Range-Free Localization Schemes for Large Scale Sensor Networks, *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, San Diego, California, USA, pp. 81-95, 2003.

[18] Y. Shang, W. Ruml, Y. Zhang, M. P. Fromherz, Localization from Mere Connectivity, *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking Computing*, Annapolis, Maryland, USA, pp. 201-212, 2003.

[19] J. S. Pan, L. Kong, T. W. Sung, P. Tsai, V. Snášel, A Clustering Scheme for Wireless Sensor Networks Based on Genetic Algorithm and Dominating Set, *Journal of Internet Technology*, vol. 19, no. 4, pp. 1111-1118, July, 2018.

[20] P. H. Namin, M. A. Tinati, Node Localization Using Particle Swarm Optimization, *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '11)*, Australia, Adelaide, pp. 288-293, December, 2011.

[21] R. Rajakumar, J. Amudhavel, Grey Wolf Optimization Algorithm for Node Localization Problem in Wireless Sensor Networks, *Journal of Computer Networks and Communications*, vol. 2017, 7348141, 2017.

[22] Q. W. Chai, S. C. Chu, J. S. Pan, A Parallel WOA with Two Communication Strategies Applied in DV-Hop Localization Method, *Journal on Wireless Communications and Networking*, vol. 2020, 50, 2020.

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

[24] S. C. Chu, H. C. Huang, J. F. Roddick, J. S. Pan, Overview of Algorithms for Swarm Intelligence, *International Conference on Computational Collective Intelligence(ICCCI 2011)*, Poland, Gdynia, pp. 28-41, 2011.

[25] J. Wu, M. Xu, F. F. Liu, M. Huang, L. H. Ma and Z. M. Lu,"Solar Wireless Sensor Network Routing Algorithm Based on Multi-Objective Particle Swarm Optimizatio", *Journal of Information Hiding and Multimedia Signal Processing*, vol. 12, no. 1, pp. 1-11, March 2021.

[26] W. Song and C. M. Huang, M. High, Average-Utility Itemsets Based on Particle Swarm Optimization, *Data Science and Pattern Recognition*, vol. 4, no. 2, pp. 19-32, 2020.

[27] A. A.Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris Hawks Optimization: Algorithm and Applications, *Future Generation Computer Systems-The International Journal of eScience*, vol. 97, pp. 849-872, 2019.

[28] D. Niculescu,B Nath, Ad Hoc Positioning System(APS) Using AOA, *IEEE INFOCOM 2003.Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies* San Francisco, CA, USA, pp. 1734-1743, 2003.

[29] H.M. Jia , C. B. Lang, D. G. Oliva, W.L. Song, X.X. Peng, Dynamic Harris Hawks Optimization with Mutation Mechanism for Satellite Image Segmentation, *Remote Sensing*, vol. 11, 1421, 2019.

[30] P. C. Song, J. S. Pan, S. C. Chu, A Parallel Compact Cuckoo Search Algorithm for Three-Dimensional Path Planning, *Applied Soft Computing*, vol. 94, 106443, 2020.

[31] Y. Hu, X. M Li, An Improvement of DV-Hop Localization Algorithm for Wireless Sensor Networks, *Telecommunication Systems*, vol. 53, pp. 13-18, 2013.

[32] X. Chen, B. Zhang, Improved DV-Hop Node Localization Algorithm in Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, vol. 8, 213980, 2012.

[33] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, Multi-Verse Optimizer: A Nature-Inspired Algorithm for Global Optimization, *Neural Computing and Applications*, vol. 27, no. 2, pp. 495-513, 2016.

[34] X. Wang, J. S. Pan, and S. C. Chu, A Parallel Multi-Verse Optimizer for Application in Multilevel Image Segmentation, *IEEE Access*, vol. 8, pp. 32018-32030, 2020.

[35] W. Niu, Z. Feng, C. Cheng, and X. Wu, A Parallel Multi-Objective Particle Swarm Optimization for Cascade Hydropower Reservoir Operation in Southwest China, *Applied Soft Computing*, vol. 70, pp. 562-575, 2018.

[36] S. Mirjalili, S. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Advances in Engineering Software*, vol. 69, no. 3, pp. 46-61, 2014.

[37] G. Song, D. Tam, Two Novel DV-Hop Localization Algorithms for Randomly Deployed Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, vol. 11, 187670, 2015.