A Hybrid Firefly with Dynamic Multi-swarm Particle Swarm Optimization for WSN Deployment

Wei-Yan Chang[1], *Prathibha Soma*[2], *Huan Chen*[1*], *Hsuan Chang*[3], *Chun-Wei Tsai*[3]

[1] *Department of Computer Science and Engineering, National Chung Hsing University, Taiwan*
[2] *Department of Information Technology, Sri Sai Ram Engineering College, India*
[3] *Department of Computer Science and Engineering, National Sun Yat-sen University, Taiwan*
keoinn@gamil.com, somaprathi25@gmail.com, huan@nchu.edu.tw, sea141498@gmail.com, cwtsai@mail.cse.nsysu.edu.tw

Abstract

Enhancing the coverage area of the sensing range with the limiting resource is a critical problem in the wireless sensor network (WSN). Mobile sensors are patched coverage holes and they also have limited energy to move in large distances. Several recent studies indicated the metaheuristic algorithms can find an acceptable deployed solution in a reasonable time, especially the PSO-based algorithm. However, the speeds of convergence of most PSO-based algorithms are too fast which will lead to the premature problem to degrade the quality of deployed performance in WSN. A hybrid metaheuristic combined with dynamic multi-swarm particle swarm optimization and firefly algorithm will be presented in this paper to find an acceptable deployed solution with the maximum coverage rate and minimum energy consumption via static and mobile sensors. Moreover, a novel switch search mechanism between sub-swarms will also be presented for the proposed algorithm to avoid fall into local optimal in early convergence process. The simulation results show that the proposed method can obtain better solutions than other PSO-based deployment algorithms compared in this paper in terms of coverage rate and energy consumption.

Keywords: Wireless sensor network, Metaheuristic algorithm, Lévy flight, Coverage rate, Energy consumption

1 Introduction

The wireless sensor has been used widely in day-to-day life because manufacturing costs are much lower and their size is small. Their features let the managers collect data, calculate, monitor, and send the information from the target area by deploying the specific type of sensors. It can help the managers make decisions according to current environment information [1]. For example, the forest ranger uses sensors that monitor the temperature and to prevent forest fires [2]. To enhance the quality of service of the application, researchers consider the coverage rate [3], energy consumption, and data transmission [4] issues to discuss. The coverage rate and energy consumption are essential issues to influence the performance of WSN. In case that the sensors deployed is not well, the WSN application might not be able to monitor some targets. This situation has a worse coverage rate for WSN, and thus some target areas cannot be monitored, called the coverage hole [5], as shown in Figure 1. Moving the mobile sensors to cover these coverage holes provide an alternative way to help us to deal with this problem. That is why we investigated some research and then attempted to develop an effective algorithm to further address this problem that hopes to use mobile sensors to patch coverage holes to increase the coverage rate of WSN. The problem of deployed mobile sensors has been proven to be NP-hard [6].

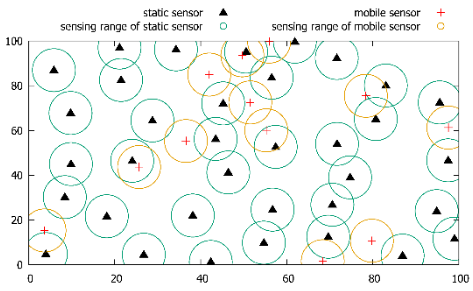


Figure 1. The sensors deployed in not well pattern

Some recent studies [7-8] adopt metaheuristics to solve the WSN problems, and they obtain better results than traditional algorithms. Although the capability of PSO [7] for WSN problems is very well, it still has the premature problem. In other words, the search diversity of most PSO-based algorithms will decrease during the convergence process. This study aims to improve the defects of PSO-based algorithms that converge fast and lack search diversity. It means that the basic idea of the proposed algorithm is to increase the search diversity of the PSO during the convergence process to keep its search from falling into a local optimum too early. The main contributions of this paper are shown in the following:

- This paper proposes a novel PSO-based algorithm, a hybrid firefly with dynamic multi-swarm particle swarm optimization (HFDMPSO), for patching coverage holes in WSN. On the other hand, it also designs a novel method to exchange information for sub-swarms that can share the best particles to increase search diversity.

*Corresponding Author: Huan Chen; E-mail: huan@nchu.edu.tw
DOI: 10.53106/160792642023072404001

- To deal with the converge fast issue, the proposed method adds the multi-swarm mechanism, which divides particles into several groups and avoids the particles move fast as they converge in the early search stage.
- To escape the local optimal, the proposed method combines the firefly algorithm (FFA) for local search and uses Le´vy flight to escape the local optimal.

To achieve these objectives, the paper is structured in the following ways. Section 2 investigates the relationship between the coverage rate and energy consumption in WSN, problem definition, and the recently PSO-based algorithm. Section 3 introduces the concept and detail of the hybrid firefly with dynamic multi-swarm particle swarm optimization. Experiments are then presented with a thorough description of the results in Section 4. Finally, results are discussed and conclusions are drawn in Section V.

2 Related Work

2.1 Problem Definition

To deal with the sensing range overlapping and coverage hole problems when the sensor was deployed, adjustable sensors are used to adjust the power of sensors in [9-10]. With the development of technology, the sensors can now be combined with vehicles or drones to become mobile sensors [11-12]. Mahboubi et al. [13] use a limited number of mobile sensors to patch coverage holes. In this study, the simulation defines the target area containing static and mobile sensors when deployed, and the mobile sensor moves to the coverage holes to patch, which will consume the energy according to moving distances. Considering the objective functions, coverage rate and energy consumption [7], it is reformed into a single equation, as shown in Eq. (1).

$$f(\phi)=\arg\max_{\phi}\left(\frac{\alpha}{\varphi}+(1-\alpha)\eta\right), \quad \textbf{(1)}$$

where ϕ represents the optimal solution for the coverage hole patching problem, φ is the energy consumption rate for solution ϕ that can be calculated by Eq. (2), η is the coverage rate for solution ϕ that can be calculated by Eq. (3), α is a control parameter that can adjust the objective function emphasizes the energy consumption or coverage rate.

$$\varphi=\sum_{i=1}^{g}(\sqrt{(x_o^i-x_\phi^i)^2+(y_o^i-y_\phi^i)^2}, \quad \textbf{(2)}$$

where g is the number of mobile sensors, x_o and y_o represents the coordinate of the mobile sensor at initial, x_ϕ and y_ϕ is the coordinate of the mobile sensor for solution ϕ.

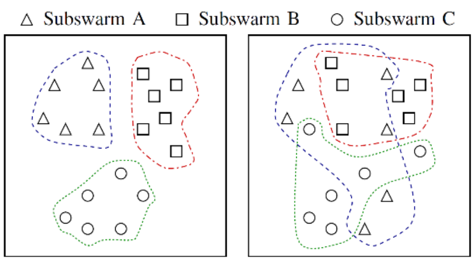
$$\eta=\frac{\sum_{P\grave{o}\rho}p(S,P)}{\rho}, \quad \textbf{(3)}$$

in Eq. (3), ρ is the target area, S represents the set of sensors, and P is an unit of target areas. Note that $p(\cdot)$ is used to judge the unit of the target area is covered by any sensor. Assume the target area unit is covered by any sensors, $p(\cdot)$ is given 1; otherwise, it is 0.

2.2 PSO-based Algorithms for WSN Deployment

The particle swarm optimization (PSO) [14] is designed according to the foraging behavior of the bird, and it used swarm intelligence to find the approximation solution for the optimization problem. In PSO, a particle represents a solution, and each particle has its own velocity. The velocity will be influenced by three forces. For the inertia velocity, the momentum of the personal best solution and for the speed of the global best solution is to update the velocity and location of particles. In the studies [15-17], Wang et al. used the PSO to solve the clustering problem of WSN that shows the possibility and potential of PSO in this research topic. Wang et al. [7] use PSO to obtain the mobile sensor move pattern that can fill the coverage hole when the static sensor is deployed. However, the particles will be easy to affected by the global best solution. The fully informed particle swarm (FIPS) [18] calculates the average of the personal best solution for each iteration and replaces the g^i item to avoid the velocity particle move too fast because of the outlier of the swarm. The FIPS solves the issue that the movement of particles is too greedy and fast from the swarm perspective. The FIPS has a certain probability of making the swarm search in regions with lower potential, which can not improve the quality of the solution. Wang et al. [19] proposed the dynamic tournament topology particle swarm optimizer (DTTPSO), which deals with these issues from the particle perspective. The result of DTTPSO shows that PSO with a grid partition is an effective and efficient method for the sensor deployment problem of WSN. It will also be able to decrease the energy that each mobile sensor uses in moving its location so as to use energy to do what the sensor is supposed to do.

Liang and Suganthan proposed the dynamic multi-swarm particle swarm optimizer (DMSPSO) [20] that used two strategies to find out a better sensor deployment plan than PSO for WSN deployment, as shown in Figure 2. The first strategy is to use a small size swarm that divides the population into several swarms to increase the search diversity of the algorithm while slowing down the convergence speed. The second strategy is the random regrouping method to deal with situations when the population size is slight lead to quickly encounters converging in the early search stage. The random regrouping method will regroup the particle of each swarm randomly at a particular period defined by the user. DMSPSO can exchange the information between different swarms and enhance the quality of the solution. The result shows that DMSPSO can provide a better deployment plan than other PSO-based algorithms.

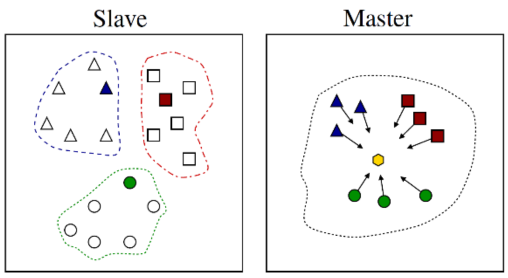


(a) (b)

(a) Sub-swarm in solution space after the initial stage (b) Sub-swarm after running out DMSPSO

Figure 2. The basic idea for DMSPSO

Niu et al. presented the multi-swarm cooperative particle swarm optimizer (MCPSO) in the study of [21] that is based on a master-slave model, as shown in Figure 3. Figure 3(a) is shown each slave swarm contains itself best particle that is filled with colors. Figure 3(b) illustrates that the master swarm collects the personal best particle of the slave swarm for each iteration and finds the global best solution in the master sarwam. The global best solution uses the polygon symbol to represent in this figure. In other words, one master swarm and several slave swarms are used in MCPSO. The implementation of each slave swarm is just like the PSO, and multiple slave swarms in this algorithm will increase the ability of global search and diversity. The master swarm will use the information of slave swarms and its own information to update its global best solution. By using this method, the particles of the master swarm can be enhanced by all the slave swarms. In other words, the master swarm will integrate information of all slave swarms, and the best particle of all the swarms can lead the search directions of the algorithm by updating its information to the master swarm.



(a) (b)

(a) Slave swarm (b) Master swarm

Figure 3. The basic concept of the MCPSO

To improve the shortcomings of PSO, which convergence too fast, Aydilek et al. [22] combined the firefly algorithm into PSO, and it developed the search process into two types. When the algorithm detects the search diversity is not enough, it will adopt the vanilla PSO to find a new solution. On the other hand, the algorithm will detect updating status of the global best solution. If the global best solution does not update in the current iteration, the search process will use the firefly algorithm to move the solution to the new position compulsorily. The search process decides to use the firefly algorithm to update the position of particles, and it only considers the objective value and distance for each particle. Let the particle with a high objective value attract nearby particle to close, just like the firefly with high brightness attracts heterosexual individuals close easily.

The different parameter combinations of PSO will generate different search results. How to choose the appropriate parameter sets or avoiding the users set the parameter by experience is an important topic. Kiran [23] used Gaussian distribution to improve the PSO update mechanism and reduce the above constants which need to be set. It uses the mean and standard deviation of the individual particle, personal best solution, and global best solution to decide the location of the distribution. The new position of particles uses the above distribution to calculate and update. Liu et al. [24] have the same opinion about the parameters combination will influence the search results of PSO. When one of the weights occupies a larger ratio, the global search ability or convergence rate will be affected. That is why Liu et al. in [24] merged the velocity and position equations and divided them into two categories: cognition-only update mechanism and social-only update mechanism. It also considers the dimension information as an additional force, that is, dimension update mechanism. This mechanism can provide the search process has more diversity. Liu et al. proposed two strategies that are time hierarchical simple particle swarm optimizer (THSPSO) and probability hierarchical simple particle swarm optimizer (PHSPSO), respectively. The main difference between them is that the THSPSO switch update mechanism according to iterations time, PHSPSO uses the random value from the uniform distribution to select a suitable mechanism to find the optimal solution.

3 Proposed Method

The notation presented in Table 1 is used throughout this paper to simplify the discussion of the proposed algorithm.

Table 1. Notations

Symbol	Description
P	The set of particles.
m	The number of sub-swarms.
R	The timer of the search switching mechanism.
τ	The threshold of the swarm regrouping mechanism.
ξ	A random number used to decide if the algorithm will run the swarm regrouping mechanism.
x	The flag used to indicate if the algorithm chooses EDMPSO or EFFA.
f	A binary number used to indicate a set of objective values of each particle.
t	A value indicates the current iteration.
t_m	The maximum number of iterations to run the algorithm.
g_b	The best solution of HFDMPSO found in the last iteration.
g_b^t	The global best soultion of particle i at iteration t.

v_i^t	The velocity of particle i at iteraton t.
ω	The inertia weight of velocity.
c_1	The acceleration factor of personal best solution.
c_2	The acceleration factor of global best solution.
c_3	The acceleration factor of DMPSO of global best solution.
r_1, r_2, r_3	Random numbers in the range [0, 1].
s	The value of a random walk.
v	The value from the normal distribution $N(0, 1)$.
μ	The value from the normal distribution $N(0, \sigma^2)$.
β	A normally distributed random number in the range [1, 2].
(x_i, y_i)	The coordinate of mobile sensor i.
$r_{i,j}$	The distance between firefly i and firefly j.
B_0	The coefficient of attraction between fireflies.
γ	The parameter that controls distance and attraction between fireflies.
$p_{b,i}^t$	The personal best solution of particle i at iteration t.

3.1 Hybrid Firefly and Dynamic Multi-Swarm Particle Swarm Optimization (HFDMPSO)

This study was inspired by HFPSO [22] and the multi-swarm concept [20] to develop a novel hybrid framework according to the quality of the global best solution and searches time to switch search mechanism. A hybrid firefly with dynamic multi-swarm particle swarm optimization (HFDMPSO) was proposed in this paper. The basic idea of the proposed algorithm is to reserve traditional particle swarm optimization as one strategery of local search. Another key idea of the proposed algorithm is the firefly algorithm which will be executed when the global best solution did not update in the recent iteration. To enhance the capability of exchanging information for each particle, the global search strategery uses the multi-swarm concept and regroup particles or exchange the particle with the best objective value according to search time. Algorithm 1 indicates the pseudo-code of proposed method. The variable x means what search mechanisms to apply for the current search stage. The proposed method adopts and improves the enhanced dynamic multi-swarm particle swarm optimizer (EDMPSO) and the enhanced firefly algorithm (EFFA) to increase diversity. After the search process, the algorithm will evaluate all particles and find the solution with the best objective values. Before going to the next iteration, it will check the iteration time according to switched time point R to change the search mechanism. Figure 4 is the flowchart of HFDMPSO. It can divide into several parts, which are initialization, evaluation, determination, and transition. The transition contains three core methods to help the algorithm increase search diversity and jump out local optimal.

Algorithm 1. HFDMPSO

```
Input:  number of subswarm $m$, set of particles $P$, the
        time point of the switching mechanism $R$, the
        threshold of regrouping swarm $\tau$.
Output: best solution $g_b$

1  while the termination criterion not met do
2      if $x$ = TRUE then
3          $P \leftarrow$ EDMPSO($P$, $\tau$)
4      else
5          $P \leftarrow$ EFFA()
6      $f \leftarrow$ Evaluation()
7      $g_b \leftarrow$ Determination($P$, $f$)
8      If (($t$ / $t_m$) % $R$) = 0 then
9          $x \leftarrow \overline{x}$
10 return $g_b$
```

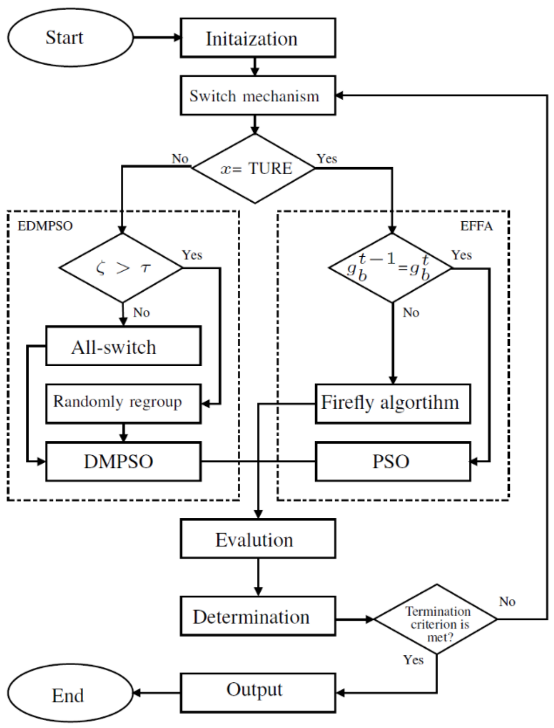


Figure 4. The flowchart of HFDMPSO

3.2 The Details of EDMPSO

To increase the diversity of the algorithm, we modified the dynamic multi-swarm particle swarm optimizer (DMSPSO) [20] and developed a novel exchange information mechanism to avoid the search process falling into local optimal, which is the all-switched method. This switching method can enforce the exchange of the search result with each sub-swarm, but it may lead to converging too fast. That is why EDMPSO retains the random grouping method of DMSPSO and modifies the velocity equation to avoid the abovementioned issue.

Algorithm 2 is the outline of EDMPSO. When the search process decides to call EDMPSO, it will obtain a random value ζ from uniform distribution and check the random value by the threshold of regrouping swarm γ to select the exchange information mechanism. When EDMPSO decides to run the all-switched method, sub-swarms randomly select a particle and exchange with other swarms forcibly, until each sub-swarm has particles that exchange from the different swarms. If it takes the randomly regrouped method, the particles in the population will stochastically assign into sub-swarms. EDMPSO also keeps the update method of the vanilla PSO algorithm as an optional mechanism for the algorithm to use. The following equation is vanilla PSO to calculate the particle's velocity and update the location of the particle.

Algorithm 2. EDMPSO

```
    Input:  the population size $k$, number of sub-swarm $m$, set of particles $P$, current
            iteration $t$, maximum iteration $t_m$, threshold of regrouping swarm $\tau$
    Output: best solution $s$
1   $\zeta \leftarrow$ Rand(0, 1)
2   if $\zeta > \tau$ then
3       /* all-switched() */
4       Use all-switch method to exchange the information
5       Each sub-swarm will get a particle from other sub-swarms
6   else
7       /* RandomlyRegroup() */
8       Use random strategy to reorganize sub-swarms
9       Randomly assign each particle to sub-swarms
10  if $t < t_m \times 0.9$ then
11      Calculate the velocity of particles by Eq. (6)
12      Update the position of particles in the solution space by Eq. (5)
13  else
14      Calculate the speed of particles by Eq. (4)
15      Update the position of particles in the solution space by Eq. (5)
16  Return $P$
```

$$\upsilon_i^{t+1} = \omega\upsilon_i^t + c_1 r_1 (p_{b,i}^t - x_i^t) + c_2 r_2 (g_{b,i}^t - x_i^t), \tag{4}$$

$$x_i^{t+1} = x_i^t + \upsilon_i^{t+1}, \tag{5}$$

where $p_{b,i}^t$ represents the personal best of the particle i, $g_{b,i}^t$ is the global best particle of swarm in iteration t, and ω is inertia weight that controls how many impacts the next iteration velocity by the particle itself velocity. c_1 and c_2 are acceleration constants for PSO. r_1 and r_2 are random number in range 0 to 1. After the exchange information stage, new sub-swarms will calculate the velocity and update position for all particles by improving Eq. (5) and Eq. (6). The improved velocity equation is shown in the following:

$$\upsilon_{i,j}^{t+1} = \omega\upsilon_{i,j}^t + s + c_3 r_3 (g_{b,j}^t - x_{i,j}^t), \tag{6}$$

$$\omega = 0.5 + \frac{1}{2(\ln(t)) + 1}, \tag{7}$$

where ω represents the adaptive weight that can obtain by Eq. (7), and s is value of Lévy flight that can calculate by Eq. (8).

$$s = \frac{1}{|v|^{1/\beta}}, \tag{8}$$

$$\sigma = \left(\frac{\Gamma(1+\beta)\cdot\sin\left(\frac{\pi\beta}{2}\right)}{\beta\Gamma\left(\frac{1+\beta}{2}\right)\cdot 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}}, \tag{9}$$

where μ is the random value from the normal distribution N $(0, \sigma^2)$, v is the random value from the normal distribution N (0, 1), β is a constant value in the range [1, 2], and σ can obtain by Eq. (9). After all the particles have calculated the new velocity, the EDMPSO will update the new location for particles by Eq. (5).

3.3 The Details of EFFA

Assume the best particle with a significant objective value, and the other particles will attract force by the best particle and modify position intensely. The search result may ignore the high quilty solution around particles that are not best in current, that search process is greedy. An improved firefly algorithm will also be used in the proposed algorithm to further enhance its performance. Algorithm 3 shows the outline of EFFA. Before the algorithm selects the method of transition solution, it detects the updating status of the global best solution. The current global best solution did not update in the last iteration, it indicates the search process has been convergence, and the solution falls into local optimal. To jump out search local optimal, the algorithm uses firefly algorithm to enforcedly change the position of particles. Otherwise, it chooses the original PSO to search in fine.

Algorithm 3. EFFA

```
    Input:  set of particles $P$, current global best solution $g_b^t$, global best solution in last
            iteration $g_b^{t-1}$
    Output: set of particles $P$
1   if $g_b^{t-1} = g_b^t$ then
2       Update the position of particles by Eq. (10) and Eq. (11)
    else
3       Calculate the speed of particles by Eq. (4)
4       Update the position of particles in the solution space by Eq. (5)
5   Return $P$
```

The firefly algorithm of the proposed algorithm will directly change the position of particles according to the objective value by Eq. (10) and Eq. (11).

$$r_{i,j} = \|X_i - X_j\| = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2}, \tag{10}$$

where $r_{i,j}$ represents the distance between particle i and j.

$$X_i = X_i + B_0 e^{-\gamma r_{i,j}^2}(X_j - X_i) + \beta, \quad (11)$$

where X represents position of particles i, B_0 is attractiveness at zero distance, and β is random value in range [1, 2].

3.4 Example

To illustrate the proposed method, we give a short example, as shown in Figure 5. Assume we have the population size is 4 and we divide it into 2 sub-swarms. Sub-swarm A contains 2 particles [5, 2, 4, 8] and [1, 2, 7, 3], and sub-swarm B has [9, 8, 3, 2] and [6, 4, 7, 1], respectively. When the proposed algorithm uses EDMPSO to run the transition stage, it will compare the random number ζ is bigger than the threshold of the regrouping swarm τ. If ζ satisfies the threshold τ, sub-swarm A randomly choices one particle to exchange with sub-swarm B. Otherwise, EDMPSO eliminates particles in sub-swarm A and sub-swarm B and randomly assigns particles into sub-swarm. Then, it according to search time to use Eq. (11) or Eq. (4) to calculate the velocity of movement and update the new position by Eq. (5) for each particle. Assume algorithm decides to use EFFA in the current time interval, it will check the updating status of the global solution firstly. If the global best solution is not updated in the last iteration, the position of particles enforced modify according to Eq. (11). Otherwise, it will choose the original PSO to calculate velocity and update the position by Eq. (4) and Eq. (5).

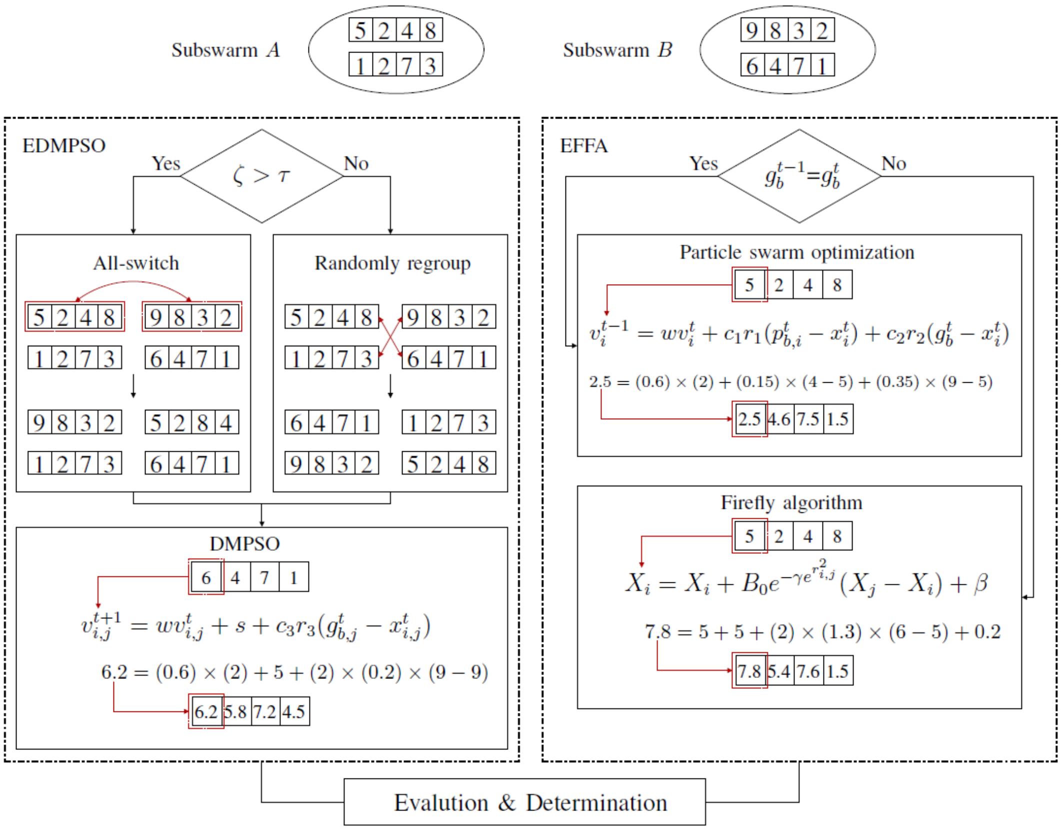

Subswarm A
5 2 4 8
1 2 7 3
Subswarm B
9 8 3 2
6 4 7 1
EDMPSO
Yes
$\zeta > \tau$
No
All-switch
Randomly regroup
DMPSO
$v_{i,j}^{t+1} = wv_{i,j}^t + s + c_3r_3(g_{b,j}^t - x_{i,j}^t)$
$6.2 = (0.6) \times (2) + 5 + (2) \times (0.2) \times (9 - 9)$
6.2 5.8 7.2 4.5
EFFA
Yes
$g_b^{t-1} = g_b^t$
No
Particle swarm optimization
$v_i^{t-1} = wv_i^t + c_1r_1(p_{b,i}^t - x_i^t) + c_2r_2(g_b^t - x_i^t)$
$2.5 = (0.6) \times (2) + (0.15) \times (4 - 5) + (0.35) \times (9 - 5)$
2.5 4.6 7.5 1.5
Firefly algorithm
$X_i = X_i + B_0 e^{-\gamma e^{r_{i,j}^2}}(X_j - X_i) + \beta$
$7.8 = 5 + 5 + (2) \times (1.3) \times (6 - 5) + 0.2$
7.8 5.4 7.6 1.5
Evalution & Determination

Figure 5. An example of HFDMPSO

4 Experimental Result

4.1 Environment & Datasets

To implement the proposed method the comparisons of algorithms are done with C++, and the simulations run on the desktop computer which equips the processor Intel i7-8700 (3.20 GHz, 6 cores, and 12 MB cache) with 16 GB memory which operating system is Windows 10. The datasets used in simulations are mainly established by referring to the configurations of [7], and the map size of the environment is 100m × 100m. The datasets contain attributes, which are the number of static sensors, the number of mobile sensors, and the sensing range, and deployment distance, as shown in Table 2.

Table 2. Configurations of datasets

Datasets	Static sensors	Mobile sensors	Sensing range	Deployment distance
DS0	35	12	5	12
DS1	35	12	8	12
DS2	35	12	8	7.5
DS3	35	8	8	12
DS4	20	12	8	12

The number of static sensors will determine the size of the initial coverage area of the target area. For example, the simulation has more static sensors, the coverage rate of the target area is greater at the initial. The simulation goal is to use the mobile sensors to raise the coverage rate as much as possible. If there are fewer mobile sensors in simulation, it means that only a few resources can be utilized to implement. The sensing range is the range that each wireless sensor can cover. The deployment distance is used to restrict the distance of static sensors at the initial deployment. This is done by keeping the deployed sensors at a certain distance. If the deployment distance is small, the overlap area will rise and the coverage rate will decrease. To conclude the above mentioned with the same number of sensors, the smaller the sensing range, the more difficult the dataset will be because the number of coverage holes will increase with the sensing area. It is a challenge for using the device to patch coverage holes and reduce the overlapped sensing area with the least possible movement.

4.2 Impacts of Parameters

The proposed method has several parameters that have to be considered in advance, which are v_L that determines the upper and lower values of the speed, the β value is the upper and lower values of Lévy flight, the threshold of regrouping swarm τ, the acceleration constant of the individually best solution c_1 in EFFA, the acceleration constant of the social best solution c_2 in EFFA, the acceleration constant c_3 of EDMPSO, and the timing to switch search mechanism R. In order to achieve better performance of the proposed method, several experiments are considered taking six parameters with dataset 0. The importance of those six parameters are listed below. The β of Lévy flight will impact the upper and lower limits of each pace, and the threshold of regrouping swarm τ will determine the probability of randomly regrouping in each iteration. R determines the switching timing parameter between EDMPSO and EFFA, and v_L will decide the max speed of each particle. The value of c_1 represents the effect of the personal best solution, the value of c_2 represents the effect of the global best solution, the value of c_3 determines the effect of swarm best solution. Figure 6 to Figure 12 show that when set value of the v_L=12.5, $\beta = 2$, $c_1 = 0.75$, $c_2 = 0.5$, $c_3 = 2$, $\tau = 10\%$ and $R = 25\%$, the proposed method can achieve the better performance.

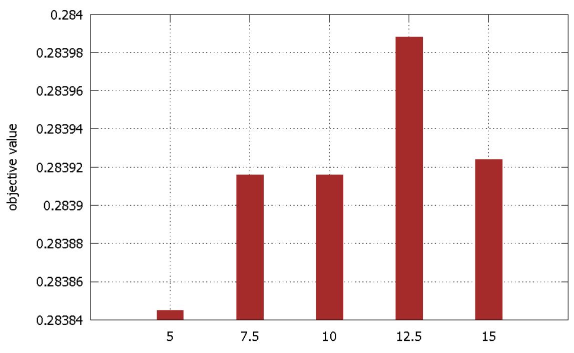


Figure 6. The impacts of parameter v_L

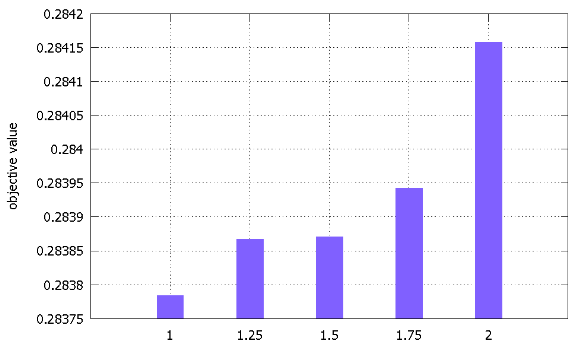


Figure 7. The impacts of parameter β

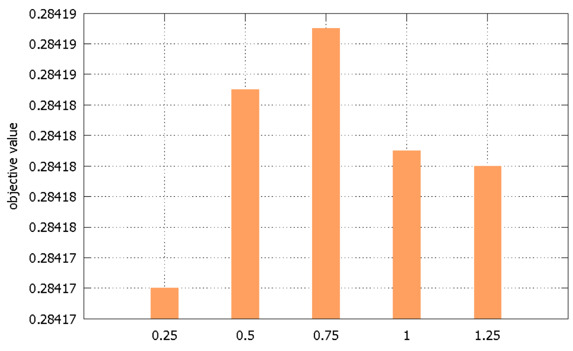


Figure 8. The impacts of parameter c_1

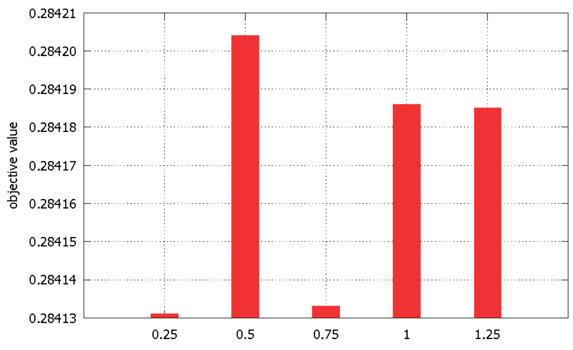


Figure 9. The impacts of parameter c_2

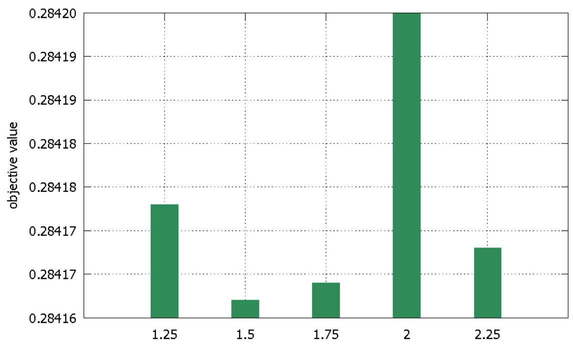


Figure 10. The impacts of parameter c_3

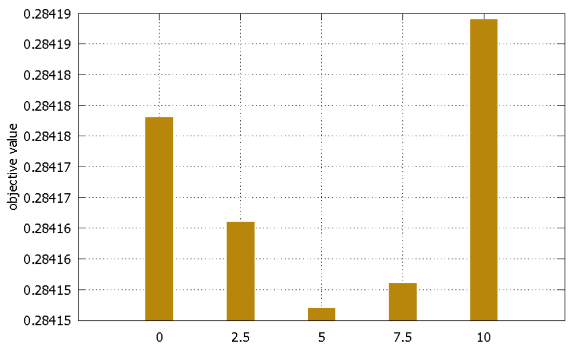


Figure 11. The impacts of parameter τ

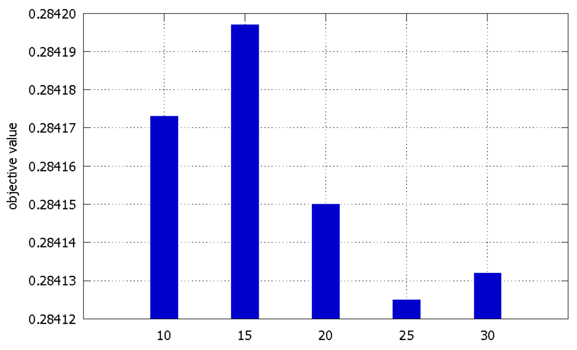


Figure 12. The impacts of parameter R%

4.3 Experimental Results

The ultimate goal of simulation is to keep the balance between coverage rate and energy consumption which uses limited mobile sensors to patch coverage holes. To evaluate the proposed method, we compared the 9 algorithms developed from PSO: vanilla PSO [14], FIPS [18], DTTPSO [19], DMSPSO [20], MCPSO [21], HFPSO [22], PSOd [23], THSPSO [24], and PHSPSO [24]. The parameters of the above algorithm are set according to its published papers. The parameters of vanilla PSO are given in the following: the number of particles is 20, ω = 1, c_1 = 2.5, and c_2 = 2.5; For DTTPSO, the number of players is 2, and it takes 6 times of tournaments; DMSPSO will regrouping sub-swarm at every 200 iterations. MCPSO has 4 slave swarms and 1 master swarm, and a swarm contains 4 particles. The c_1 and c_2 for HFPSO are set to 1.49. The t_1 and t_2 of THSPSO are set to 0.2 and 0.8, respectively. The p_1 and p_2 are respectively set to 0.6 and 0.9 in PHSPSO. The objective values are recorded for every evaluation time for each comparison algorithm, and it take 30 runs to calculate the average. Finally, the convergence plots are present, as shown in Figures 13 to Figure 16.

As shown in most datasets Figure 13 to Figure 16, the algorithm with the multi-swarm features can have better performance than other algorithms, which improved the single swarm best solution. By improving the perspective of the single swarm, the best solution may prevent the drawback of PSO, which is easy to trap in the local optima solution. However, the single swarm-based algorithms still cannot avoid all the particles as all the particles may not be impacted by the best solution of the same swarm, and it also leads to the global search ability decrease. It means that the single swarm-based algorithms influence the same global best solution, so converging too fast will lead to search diversity insufficient.

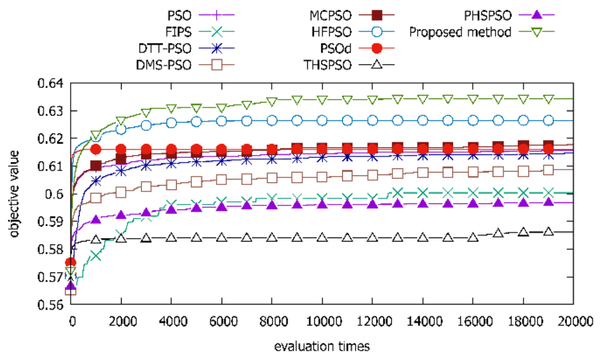


Figure 13. The results of convergence for Dataset DS1

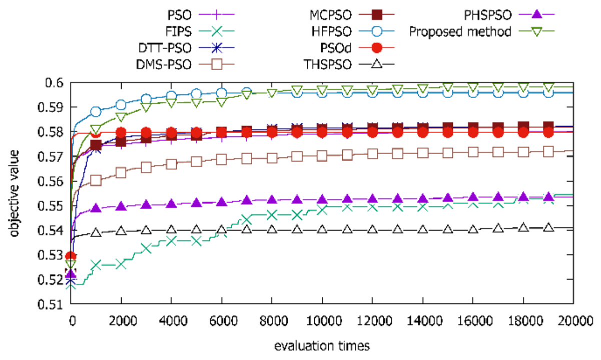


Figure 14. The results of convergence for Dataset DS2

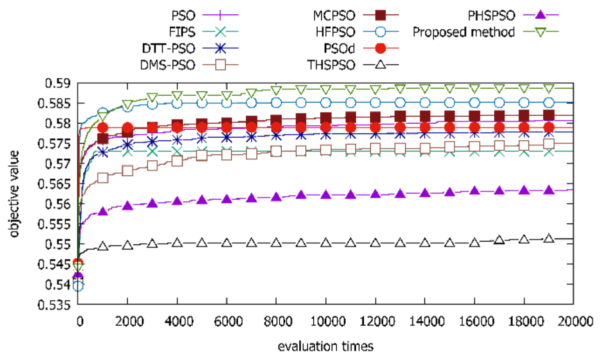


Figure 15. The results of convergence for Dataset DS3

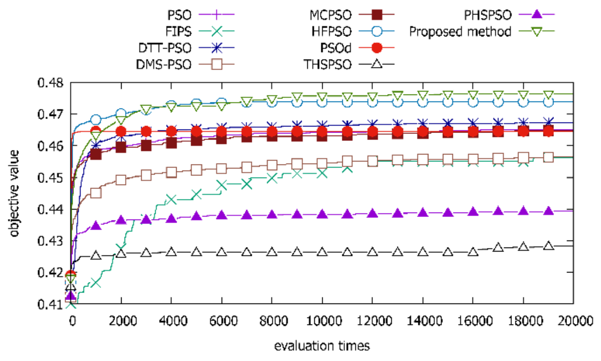


Figure 16. The results of convergence for Dataset DS4

In addition, we also sort out results into a numeric table, as shown in Table 3. The values in the tables represent the objective value in the final evaluation times for each comparison algorithm. The numeric results show that HFDMPSO can perform better than other algorithms in all datasets. It means the proposed method has a better ability to balance the coverage rate and energy consumption. While the mobile sensors move their location to offset the coverage hole areas as much as possible, the lifetime of the whole WSN also is considered simultaneously. It is worth mentioning that every 25% iteration time of the proposed method will repeatedly change the global search strategy and local search strategy. The objective value is easy to rise slightly at that point. Table 4 indicates the running time of each algorithm that performs evaluation time 20,000 times. The value of Table 4 is the average running time in 30 runs, and the unit is seconds. Although the proposed method is not the fastest, it still closes to PSO, FIPS, DTTPSO, and HFPSO. The fastest algorithm is PSOd because it directly modifies the position of particles according to the position of current, global best, and individual best. It avoids the calculated equation of velocity to reduce the running time.

Table 3. Comparison of different deployment algorithms

Algorithm	DS1	DS2	DS3	DS4
PSO	0.6156	0.5798	0.5806	0.4650
FIPS	0.6003	0.5543	0.5730	0.4563
DTTPSO	0.6148	0.5820	0.5778	0.4673
DMSPSO	0.6088	0.5721	0.5750	0.4564
MCPSO	0.6177	0.5819	0.5820	0.4650
HFPSO	0.6264	0.5958	0.5851	0.4738
PSOd	0.6161	0.5797	0.5789	0.4645
THSPSO	0.5861	0.5409	0.5512	0.4283
PHSPSO	0.5968	0.5534	0.5634	0.4393
HFDMPSO	0.6343	0.5982	0.5887	0.4763

Table 4. Experimental results in terms of the running time

Algorithm	DS1	DS2	DS3	DS5
PSO	474.75	473.76	322.02	468.83
FIPS	474.18	472.67	324.70	473.01
DTTPSO	462.56	463.14	313.26	463.32
DMSPSO	471.10	467.61	319.38	471.26
MCPSO	2268.03	2258.63	1542.53	2263.73
HFPSO	463.41	460.34	313.02	463.33
PSOd	376.59	375.40	256.69	376.80
THSPSO	960.22	956.87	650.33	958.55
PHSPSO	937.07	939.69	639.67	933.12
HFDMPSO	472.90	471.80	325.94	474.56

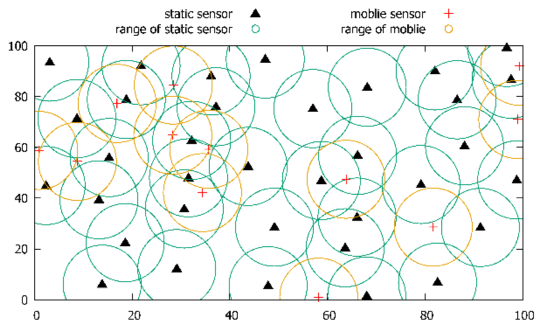
4.4 Analysis

In order to prove the efficiency of the proposed method when compared with other algorithms, we used Wilcoxon signed-rank test to evaluate the performance. The statistical analysis result is shown in Table 5. The level of statistical significance is set to 0.05, which means the result is better or worse caused by random in 5% probability. Therefore, we can claim the results with good evidence against the null hypothesis when the p-value < 0.05. The "Δ" rows of Table 5 have two types of symbols: "+" and "–", which represent the statistical significance and not statistical significance.

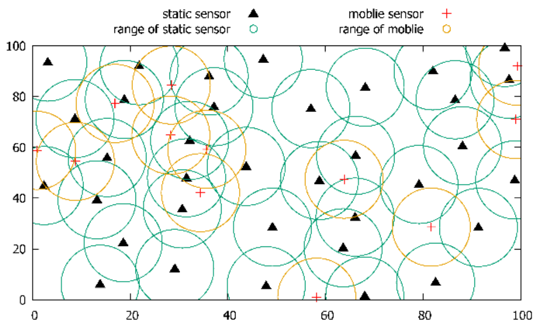
Table 5. Wilcoxon sign rank test

Algorithm		DS1	DS2	DS3	DS4
	Z-test	-4.782	-4.782	-4.700	-4.782
PSO	p-value	0.000	0.000	0.00	0.000
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.782	-4.782
FIPS	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.762	-4.782
DTTPSO	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.782	-4.782
DMSPSO	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.535	-4.782
MCPSO	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+
	Z-test	-4.186	-2.725	-2.910	-2.232
HFPSO	p-value	0.000	0.006	0.004	0.026
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.679	-4.782
PSOd	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.782	-4.782
THSPSO	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+
	Z-test	-4.782	-4.782	-4.782	-4.782
PHSPSO	p-value	0.000	0.000	0.000	0.000
	Δ	+	+	+	+

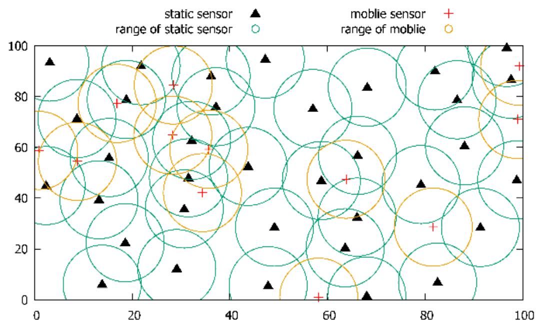
Finally, the deployed pattern of the result is visualized, and the proposed method is optimized, as shown in Figure 17. The figures on the right-hand side represent the deployed patterns found by the proposed method in the last evaluation times. The caption shows the name of the dataset in figures represents the status and location of sensors at initialization. From visualization of deployed pattern, we can observe that the proposed method can patch the coverage hole effectively.



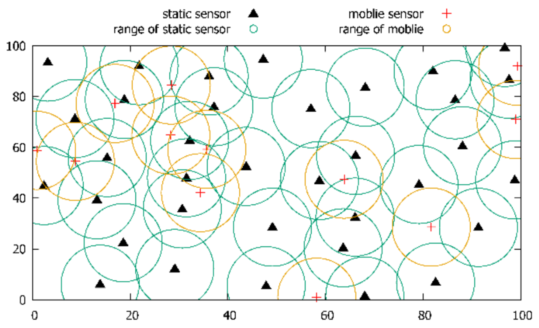
(a) The sensors distribution initialize stage for DS1



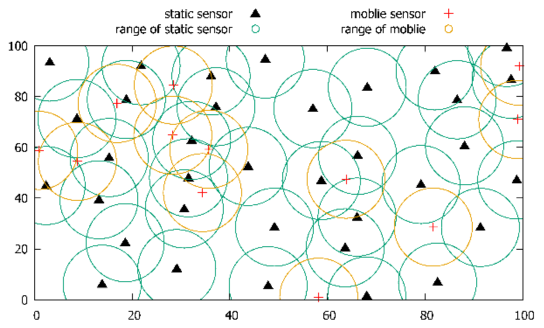
(b) The sensors distribution after run HFDMPSO for DS1



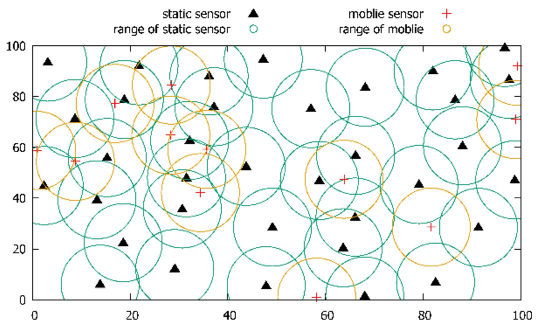
(c) The sensors distribution initialize stage for DS2



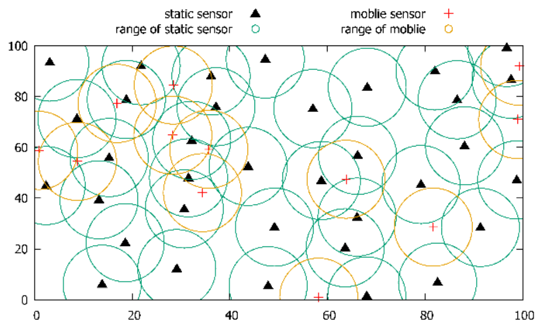
(d) The sensors distribution after run HFDMPSO for DS2



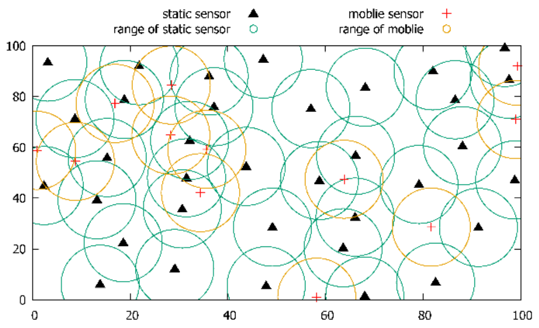
(e) The sensors distribution initialize stage for DS3



(f) The sensors distribution after run HFDMPSO for DS3



(g) The sensors distribution initialize stage for DS4



(h) The sensors distribution after run HFDMPSO for DS4

Figure 17. The comparison of visualization for sensors distribution includes initializing stage and runs the proposed method to patch coverage hole

5 Conclusion

In this paper, it is proposed a hybrid metaheuristic algorithm to deal with the sensor deployment problems in wireless sensor network. To prevent the drawback of PSO, which is easily trapped in the local optima and premature convergence in the early searching phase, the proposed method contains enhanced dynamic multi-swarm particle swarm optimizer with Lévy flight, the enhanced firefly algorithm, and according to search timing to switch suitable mechanisms. In the simulation, we compare with 9 algorithms that developed from PSO. The experimental results ow that the proposed method can obtain a better objective value than other algorithms in different environments, which represent the proposed method can effectively balance the coverage rate and energy consumption. Therefore, the wireless sensor network enhances its coverage rate and considers its entire lifetime. As a part of future work, many issues such as making the value of user-defined parameters adaptable can be given a better solution.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions on the paper. This research was partially supported in part by the National Science and Technology Council (NSTC) of Taiwan, R.O.C., under the grant numbers NSTC-110-2221-E-005-032-MY3, NSTC-111-2634-F-005-001, NSTC-111-2222-E-110-006-MY3, NSTC-112-2628-E-110-001-MY3, and NSTC-112-2634-F-110-001-MBK.

References

[1] D. Kandris, C. Nakas, D. Vomvas, G. Koulouras, Applications of wireless sensor networks: An up-to-date survey, *Applied System Innovation*, Vol. 3, No. 1, pp. 1-24, March, 2020.

[2] S. R. J. Ramson, D. J. Moni, Applications of wireless sensor networks— A survey, *Proceedings of International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology*, Coimbatore, India, 2017, pp. 325-329.

[3] A. Tripathi, H. P. Gupta, T. Dutta, R. Mishra, K. K. Shukla, S. Jit, Coverage and connectivity in WSNs: A survey, research issues and challenges, *IEEE Access*, Vol. 6, No. 1, pp. 26971-26992, June, 2018.

[4] O. O. Ogundile, A. S. Alfa, A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks, *Sensors*, Vol. 17, No. 5, pp. 1-51, May, 2017.

[5] S. M. Mohamed, H. S. Hamza, I. A. Saroit, Coverage in mobile wireless sensor networks (M-WSN): A survey, *Computer Communications*, Vol. 110, No. 1, pp. 133-150, September, 2017.

[6] N.-T. Nguyen, B.-H. Liu, The mobile sensor deployment problem and the target coverage problem in mobile wireless sensor networks are NP-hard, *IEEE Systems Journal*, Vol. 13, No. 2, pp. 1312-1315, June, 2019.

[7] J. Wang, C. Ju, H.-J. Kim, R. S. Sherratt, S. Lee, A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs, *Cluster Computing*, Vol. 22, No. suppl. 1, pp. 1787-1795, January, 2019.

[8] C.-W. Tsai, P.-W. Tsai, J.-S. Pan, H.-C. Chao, Metaheuristics for the deployment problem of WSN: A review, *Microprocessors and Microsystems*, Vol. 39, No. 8, pp. 1305-1317, November, 2015.

[9] C. Wang, B. Wang, H. Xu, W. Liu, Energy-efficient barrier coverage in WSNs with adjustable sensing ranges, *Proceeding of IEEE Vehicular Technology Conference*, Yokohama, Japan, 2012, pp. 1-5.

[10] B. Khalifa, A. M. Khedr, Z. A. Aghbari, A coverage maintenance algorithm for mobile WSNs with adjustable sensing range, *IEEE Sensors Journal*, Vol. 20, No. 3, pp. 1582-1591, February, 2020.

[11] P. D. Mitcheson, D. Boyle, G. Kkelis, D. Yates, J. A. Saenz, S. Aldhaher, E. Yeatman, Energy-autonomous sensing systems using drones, *Proceedings of IEEE Sensors*, Glasgow, UK, 2017, pp. 1-3.

[12] A. Kaushik, D. K. Lobiyal, Localization in wireless sensor networks using a mobile anchor and subordinate nodes, in: S. M. Thampi, J. Lloret Mauri, X. Fernando, R. Boppana, S. Geetha, A. Sikora (Eds.), *Applied Soft Computing and Communication Networks*, vol. 187, Springer, Singapore, 2021, pp. 177-188.

[13] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, V. Marbukh, Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors, *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 1, pp. 163-174, February, 2014.

[14] J. Kennedy, R. C. Eberhart, Particle swarm optimization, *Proceedings of International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[15] J. Wang, Y. Cao, B. Li, H.-J. Kim, S. Lee, Particle swarm optimization based clustering algorithm with mobile sink for WSNs, *Future Generation Computer Systems*, Vol. 76, pp. 452-457, November, 2017.

[16] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt, L. Wang Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs, *Computers, Materials & Continua*, Vol. 62, No. 2, pp. 695-711, 2020.

[17] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, G. Kim, A PSO based energy efficient coverage control algorithm for wireless sensor networks, *Computers, Materials & Continua*, Vol. 56, No. 3, pp. 433-446, 2018.

[18] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: Simpler, maybe better, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 204-210, June, 2004.

[19] L. Wang, B. Yang, J. Orchard, Particle swarm optimization using dynamic tournament topology, *Applied Soft Computing*, Vol. 48, pp. 584-596, November, 2016.

[20] J. J. Liang, P. N. Suganthan, Dynamic multi-swarm particle swarm optimizer, *Proceedings of IEEE Swarm Intelligence Symposium*, Pasadena, CA, USA, 2005, pp. 124-129.

[21] B. Niu, Y. Zhu, X. He, H. Wu, MCPSO: A multi-swarm cooperative particle swarm optimizer, *Applied Mathematics and Computation*, Vol. 185, No. 2, pp. 1050-1062, February, 2007.

[22] I. B. Aydilek, A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, *Applied Soft Computing*, Vol. 66, pp. 232-249, May, 2018.

[23] M. S. Kiran, Particle swarm optimization with a new update mechanism, *Applied Soft Computing*, Vol. 60, pp. 670-678, November, 2017.

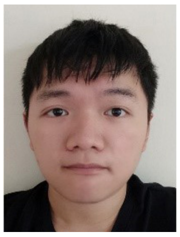
[24] H.-R. Liu, J.-C. Cui, Z.-D. Lu, D.-Y. Liu, Y.-J. Deng, A hierarchical simple particle swarm optimization with mean dimensional information, *Applied Soft Computing*, Vol. 76, pp. 712-725, March, 2019.

Biographies

Wei-Yan Chang is a Ph.D. student with department of computer science and engineering, National Chung Hsing University, Taiwan. He received B.S. and M.S. degree from the National Formosa University, Taiwan, in 2014 and 2018, respectively.

Prathibha Soma received the B.E. degree in Computer Science and Engineering from Gulbarga University, in 1997. And M.E. degree in Computer Science and Engineering from Sathyabama Univerisy, in 2008. And Ph.D. in Cloud Computing from Anna University in 2019.

Huan Chen is an associate professor with the department of computer science and engineering, National Chung Hsing University, Taiwan. He received the B.S. and M.S. degrees from the National Tsing Hua University, Taiwan, in 1993 and 1995, respectively, and Ph.D. degree from the University of Southern California, USA, in 2002.



Hsuan Chang was a master's student with department of computer science and engineering, National Sun Yat-sen University, Taiwan. He received B.S degree from National Dong Hwa University, Taiwan, in 2018.

Chun-Wei Tsai received his Ph.D. degree in Computer Science and Engineering from National Sun Yat-sen University, Kaohsiung, Taiwan in 2009, where he is currently an assistant professor. His research interests include computational intelligence, data mining, cloud computing, and internet of things.