Nat Comput (2017) 16:5–13
DOI 10.1007/s11047-015-9519-0

An improved dynamic deployment method for wireless sensor network based on multi-swarm particle swarm optimization

Qingjian Ni[1] · Huimin Du[2] · Qianqian Pan[3] · Cen Cao[1] · Yuqing Zhai[1]

Published online: 14 September 2015

Abstract Dynamic deployment methods for wireless sensor network (WSN) can improve the quality of service (QoS) of the network by adjusting positions of mobile nodes. In the dynamic deployment problem model of this paper, not only the coverage rate of WSN but also the moving distance of mobile nodes is taken into consideration. This kind of model can be abstracted into multi-objective optimization problem, and particle swarm optimization (PSO) is introduced to solve this problem. In this paper, combined with previous work, an improved dynamic deployment method is proposed based on multi-swarm PSO. Specifically, we propose a discrete PSO to calculate the distance of mobile solutions, and a multi-swarm PSO is designed to optimize network performance for enhancing the QoS of deployment which includes higher coverage rate and lower energy consumption of mobile nodes. Experimental results demonstrate that the proposed method has a good performance in solving the WSN deployment problem.

Keywords Dynamic deployment · Discrete particle swarm optimization · Multi-swarm particle swarm optimization

✉ Qingjian Ni
nqj@seu.edu.cn

[1] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China

[2] College of Software Engineering, Southeast University, Nanjing 211189, China

[3] School of Information Science and Engineering, Southeast University, Nanjing 211189, China

1 Introduction

Node deployment is an important research area in wireless sensor network, and the coverage quality directly affects the QoS of the network. In WSN, the data acquisition nodes are often composed of static and dynamic nodes. Coverage quality of static nodes is generally depending on the initial deployment strategy. However, dynamic nodes with moving ability can change their geographic topology to enhance the coverage quality of the entire network. The initial positions of nodes are always random with low coverage, where the mobility of nodes can improve the situation, realizing self-organization of the network.

Researchers have carried out in-depth exploration of this area, and the main indicator concerned is the coverage rate of network. Aleksandra and Gavrilovska (2011) designed a distributed algorithm to improve the connectivity and coverage of network, in which the space was organized to hexagonal grid, and cluster heads were determined in the center for each grid cell. Then nodes inside and adjacent to the cells could be rearranged to improve the performance. A virtual force algorithm was proposed by Zou and Chakrabarty (2003) as a node deployment method to maximize the coverage rate, where the cluster heads instructed the movement of the nodes around. When facing with large-scope situations, Ma and Yang (2007) presented an adaptive triangular deployment algorithm for large-scale mobile sensor networks, this strategy adapted node deployment to regular triangles and a sector-based detection was utilized. Wang et al. (2007) combined the virtual force algorithm with PSO, proposed an improved dynamic deployment algorithm, which performs better than many other methods when dealing with probabilistic detection model. Aziz et al. (2007) used Voronoi diagram to evaluate the solution, and chose PSO to find the optimal deployment plan. Compared with grid point deployment, Voronoi diagram provided an easier implementation and the result was considerable.

The topology of sensors is dynamic since those mobile nodes can adjust their positions during their life cycle. In this case, optimization method can be carried out over and over again (once node failure occurs or environment changes) to help deploy the whole network.

In most previous research of node deployment, the moving distance of mobile nodes is not concerned too much. It is often assumed that mobile nodes can reach any place in the area and the energy consumption for moving is not considered. In fact, this metric is rather crucial since it decides the distribution of energy consumption: if the moving distance of nodes is shorter, less energy will be consumed. Furthermore, a shorter moving distance leads to a quicker deployment (fast convergence) since the mobile sensors can reach their target positions as soon as possible. In this paper, the moving distance is considered as an important indicator in the process, which is directly related to the total energy that all dynamic nodes take to move to target locations. Due to the energy limitation, a lower and more stable energy cost strategy can lead to a higher quality of the network.

Furthermore, a multi-swarm PSO (MPSO) with various PSO variants is proposed to solve the problem of dynamic deployment. Multi-swarm strategies in evolutionary algorithm have been performing excellently, and in general, multi-swarm PSO implements the same evolution strategy in each population. Blackwell and Branke (2004) proposed the multi-swarm optimization method based on charged PSO and took some additional methods to prevent the situation of balanced attractors, and the charged particles and charged quantum particles act in different rules between different swarms. Liang and Suganthan (2005) put forward a dynamic multi-swarm PSO by regrouping the swarms during the process using various regrouping schedules, where each swarm maintains a small number of particles to achieve a higher diversity. Solomon et al. (2011) implemented a collaborative multi-swarm PSO to achieve high parallelism to test the suitability of GPU in a distributed computing environment. This multi-swarm method with swapping operations among the swarms and repulsion factor, which is stated in Vanneschi et al. (2010), is used to ensure a high parallelism and good performance. Liu et al. (2011) designed a modified multi-swarm PSO where the sub-swarms are scheduled by the multi-swarm scheduling module and there are four rules for the scheduler to maintain the swarms including information transmits and other policies, and the algorithm is integrated with Support Vector Machine to do feature selection jobs. Zhao et al. (2011) improved the dynamic multi-swarm PSO by hybridizing it with harmony search and developed Dynamic Multi-Swarm Particle Swarm Optimizer with Harmony Search (DMS-PSO-HS), where every sub-swarm also acts as a harmony search population to speed up the convergence and the swarms are regrouped frequently as well. Mukhopadhyay and Banerjee (2012) proposed Chaotic Multi Swarm Particle Swarm Optimization (CMS-PSO) where the diversity of generic PSO is improved with the help of chaotic sequence, by injecting the chaotic system parameters into the update equations of PSO and enhanced the performance. A Finder–tracker (FTPSO) multi-swarm PSO was proposed by Yazdani et al. (2013) where the method is used to find peak values in searching space and track them when the environment changes. In this algorithm there are several tracker swarms and one finder swarm, and the global optimal is retained in one of the sub-swarms.

Since multi-swarm strategy can perform better than the single swarm method, it is taken into use to solve the complex problem in WSN deployment. While in this paper, an improved multi-swarm PSO is proposed with different PSO evolutionary methods, i.e., a heterogeneous MPSO. Three different particle swarm optimizers are combined to reach high performance, which is different from the existed multi-swarm methods above where only one kind of method is implemented among the swarms, and will be discussed later.

The remainder of this paper, which is an substantial extension of the work (Du et al. 2014), is organized as follows. Section 2 describes several PSO variants which will be used in a multi-swarm PSO proposed in the later section. In Sect. 3, the optimization problem model for WSN dynamic deployment and the detection model are described, and the method for the WSN dynamic deployment is proposed based on the multi-swarm PSO. The experiment and analysis are given in Sect. 4. Finally, Sect. 5 gives the conclusion and the future work.

2 Particle swarm optimization variants

This section describes several variants of PSO algorithm. These PSO variants are used in a multi-swarm PSO proposed in this paper.

2.1 PSO with inertia weight

Shi and Eberhart (1998) initially introduced the inertia weight to improve the traditional PSO algorithm. The inertia weight represents the speed inertia of the particles in population. In general, high weight values can make population search more freely, and after reaching a certain degree, low weight values are conducive to the local exploitation. The calculation of velocity and position in

PSO algorithm with inertia weight ω is given in Eqs. (1) and (2).

$$v_i^{(t+1)} = \omega v_i^{(t)} + r_1 c_1 (p_i^{(t)} - x_i^{(t)}) + r_2 c_2 (p_c^{(t)} - x_i^{(t)}) \tag{1}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{2}$$

where p_i is individual's optimal solution, and p_c is the optimal solution in the whole population. There are some changing strategies of the inertia weight, like decreasing the value from 0.9 to 0.1 through the evolution. Such strategies make the population have a strong exploration capability at the initial stage and a strong exploitation capability in later iterations.

2.2 PSO with constriction factor

PSO with constriction factor was firstly introduced by Clerc and Kennedy (2002). This PSO variant is another way to control the situation of particles in population. After the particles reach a dominant region, the inhibition of the constriction factor on population concussion can lead to a fast convergence of the whole population. The position update equation is the same with Eq. (2) and the velocity update equation is shown in Eq. (3), where χ may take a value of 0.729.

$$v_i^{(t+1)} = \chi\left\{v_i^{(t)} + r_1 c_1 \left(p_i^{(t)} - x_i^{(t)}\right) + r_2 c_2 \left(p_c^{(t)} - x_i^{(t)}\right)\right\} \tag{3}$$

2.3 Dynamic probabilistic PSO

Kennedy (2005) firstly introduced a kind of PSO without velocity. Ni and Deng (2011) systematically integrated this kind of PSO variant: Dynamic probabilistic PSO (DPPSO). The core equations of DPPSO are given by Eqs. (4) and (5).

$$CT_{id}(t) = \sum_{k=1}^{K} P_{kd}(t)/K - X_{id}(t) \tag{4}$$

$$OT_{id}(t) = \sum_{k=1}^{K} |P_{id} - P_{kd}|/K \tag{5}$$

where t is the number of iteration of the current evolution, i is the index of a particle, k is the index of neighborhood particles, K is the number of particles in the neighborhood. P_{kd} is the individual optimal position of neighborhood particle whose index is k, d is the dimension index of particle i. Position update equation is shown in Eq. (6).

$$\begin{aligned} X_i(t+1) &= X_i(t) + \alpha(X_i(t) - X_i(t-1)) \\ &\quad + \beta CT_i(t) + \gamma Gen() OT(t) \end{aligned} \tag{6}$$

where $Gen()$ is a random number generator, which is often set to be a distribution function satisfying a specific random distribution. α, β, γ are the weights.

$Gen()$ is very important to DPPSO, and it has a direct impact on the population sampling method for the next generation. Different DPPSO algorithms have different features (Ni and Deng 2011): DPPSO-Gaussian has fast convergence in the early iterations; DPPSO-Logistic and DPPSO-Hyperbolic Secant have a strong exploration capability in the later stage of evolution, which ensures the particles have a strong ability to escape from local optima; DPPSO-Cauchy has a good performance on a small number of benchmark functions, indicating DPPSO-Cauchy is suitable for solving similar engineering problems.

In this paper, we choose DPPSO-Gaussian algorithm, and the Gaussian equation is as Eq. (7).

$$f(x) = ae^{-(x-b)^2/c^2} \tag{7}$$

where x is a random number. Based on our previous experiments and investigation, x is set between $[-2, 2]$, and $a = 1.73, b = 0, c = 1.41$.

3 Problem model and proposed method

3.1 Optimization problem model for WSN dynamic deployment

In WSN dynamic deployment problem, two main factors to be measured are coverage rate and moving distance of the network, which are thus two optimization targets for the candidate solutions of the algorithm.

Coverage rate C : Abstract the region of interest (ROI) to a set of points (grid points or random covering points), and the coverage rate is the proportion of the points that are effectively covered in the ROI. This principle is also described in literature (Kulkarni and Venayagamoorthy 2011).

Moving distance d : The moving distance of a node is the distance between its initial position and its moving target, while for the network, the total moving distance is the sum of all the moving distances of mobile nodes.

The coverage rate is the predominant factor since a short moving distance makes no sense without a high coverage rate. The fitness function is noted as f, and it is supposed that a higher coverage rate and/or a shorter moving distance contribute to a better fitness value. Fitness function is designed as Eq. (8) according to those features.

$$f = \left(\frac{C}{Th}\right)^m - \frac{\hat{d}}{S} \tag{8}$$

where Th is a threshold value of coverage rate C, and if the difference of C of two solutions is over Th, then C will become the predominant indicator. Assume that $C > Th$, m is a parameter associated with the statistic value of moving distance and $m > 1$. $\hat{d}$ is the average moving distance of all mobile nodes in the network, S is the region width. In this paper, the parameters are set as follows: $Th = 5\,\%, m = 1.4$.

According to the fitness function, it can be found that this is an NP-hard problem and it is not possible to find the theoretically optimum value with reasonable time. Thus PSO is suitable to be used to obtain the solutions that could be accepted in practice.

3.2 Detection models for WSN

There are two detection models for WSN coverage problem: the binary detection model and the probabilistic detection model (Zou and Chakrabarty 2003). The binary detection model has a wide range of applications while the probabilistic detection model provides a more realistic measure strategy. Presume a set of sensors $S = \{s_1, \ldots, s_n\}$, the probabilistic for a point $P(x, y)$ to be covered by s_i is noted as $c_{xy}(s_i)$, which is described in Eq. (9).

$$c_{xy} = \begin{cases} 0 & if \quad R_s + R_e \le d(s_i, P) \\ e^{-\lambda\alpha^{\beta}} & if \quad R_s - R_e < d(s_i, P) < R_s + R_e \\ 1 & if \quad d(s_i, P) \le R_s - R_e \end{cases} \tag{9}$$

where R_e is the uncertainty factor in detection process. It can be found that when $R_e = 0$, Eq. (9) acts as the binary model, otherwise it turns to the probabilistic one.

Li et al. (2005) proposed an improved probabilistic model to reach a more precise simulation result, which is given by Eq. (10).

$$c_{xy} = \begin{cases} 0 & if \quad R_s + R_e \le d(s_i, P) \\ e^{-\alpha_1\lambda_1^{\beta_1}/\lambda_2^{\beta_2}+\alpha_2} & if \quad R_s - R_e < d(s_i, P) < R_s + R_e \\ 1 & if \quad d(s_i, P) \le R_s - R_e \end{cases} \tag{10}$$

where $\lambda_1 = R_e - R_s + d(s_i, P)$, $\lambda_2 = R_e + R_c - d(s_i, P)$. $\alpha_1, \alpha_2, \beta_1, \beta_2$ are the property parameters determined by the physical features of different sensor nodes. The probability for $P(x, y)$ to be covered by a set of sensors S_{ov} can be presented as Eq. (11).

$$c_{xy}(S_{ov}) = 1 - \prod_{s_i \in S_{ov}} (1 - c_{xy}(s_i)) \tag{11}$$

and P is covered when it meets the condition given by Eq. (12).

$$c_{xy}(S_{ov}) \ge C_{th} \tag{12}$$

where C_{th} is the cover probability threshold.

Experiments in this paper take both binary detection model and probabilistic detection model into account.

3.3 The proposed PSO algorithms

3.3.1 *Discrete PSO for the computation of moving distance in WSN*

Traditional PSO algorithms are mainly used to solve continuous problems, while Clerc (2004), Wang et al. (2003) and Shi et al. (2007) proposed several discrete strategies for PSO when solving discrete problems such as traveling salesman problem. When dealing with the computation of moving distance, an improved discrete PSO is introduced in this paper. New operators are designed to discretize traditional PSO, and the equations are as Eqs. (13)–(15).

$$\begin{aligned} V_i^{(t+1)} = {} & W(\omega) \otimes V_i^{(t)} \circ W(c_1 r_1) \otimes \left(P_{id} \ominus X_i^{(t)}\right) \\ & \circ W(c_2 r_2) \otimes \left(P_{gd} \ominus X_i^{(t)}\right) \end{aligned} \tag{13}$$

$$X^{(t+1)} = X^{(t)} \oplus V^{(t)} = \left\{x_1^{(t)} + \ldots x_N^{(t)}\right\} \oplus \left\{v_1^{(t)}, \ldots, v_N^{(t)}\right\} \tag{14}$$

where W is a normalizing function to produce weighting coefficient. Assume there are k factors in all ($k = 3$ in (13)) for an equation, then W is built as

$$W(factor_i) = \frac{factor_i}{\sum_{i=0}^{k} factor_i} \tag{15}$$

Operators are defined as follows.

$\otimes$ ***Multiplication operator for coefficient and velocity:*** The result is a vector containing **null** values. The coefficient stands for the retention proportion of this vector and the retained index should be unique among all the vectors to be added.

$\circ$ ***Addition operator for two velocities:*** Merge two vectors to a new vector according to the item indexes. For example, $\{v_{i1}, v_{i2}, \mathbf{null}, v_{i4}\} \circ \{\mathbf{null}, \mathbf{null}, v_{i3}, \mathbf{null}\} = \{v_{i1}, v_{i2}, v_{i3}, v_{i4}\}$. According to Eq. (15) and the multiple method above, all the sub-vectors will be merged to exactly one complete vector.

$\ominus$ ***Subtraction operator for two positions:*** For each dimension, define operate mode for each dimension as

$$p_d \ominus x_d = \begin{cases} p_d, & if \quad rand() \ge \alpha \\ x_d + \gamma(p_d - x_d), & if \quad \alpha > rand() > \beta \\ x_d, & otherwise \end{cases} \tag{16}$$

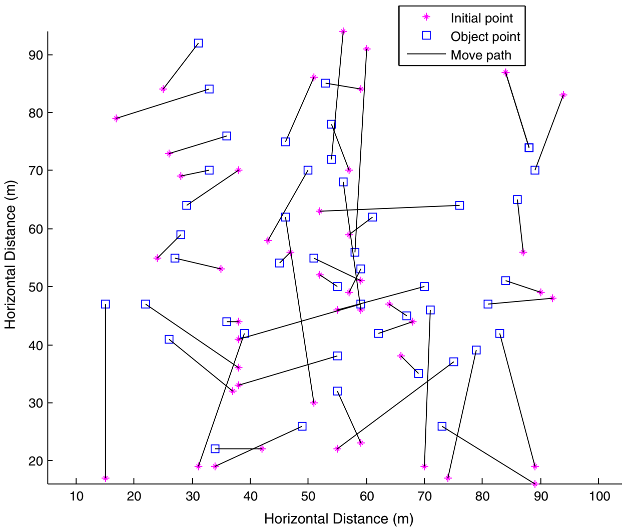

Fig. 1 Schematic diagram of moving planning. (Color figure online)

where $0<\gamma<1, 0<\beta\leq\alpha<1, rand()\in[0,1]$, and (β,α) is the probability interval of interference factor, which is optional.

$\oplus$ ***Addition operator for a velocity and a position:*** Always assume that velocity and position vectors are of the same dimension. Operations for each dimension are defined as

$$x_d \oplus v_d = \begin{cases} v_d, & if \quad rand() \geq \alpha' \\ x_d + \gamma'(v_d - x_d), & if \quad \alpha' > rand() > \beta' \\ x_d, & otherwise \end{cases} \tag{17}$$

THEN do $\quad x_i \leftarrow x_d \quad$ *where* $\quad x_i = x_d \oplus v_d$.

where $0<\gamma'<1, 0<\beta'\leq\alpha'<1, rand()\in[0,1]$, and (β',α') is the probability interval of interference factor which is optional. This operation has two steps, calculating $x_d \oplus v_d$ and then swapping the result with x_d inside the vector.

Figure 1 is a schematic diagram of a moving plan which is given by the proposed algorithm. In a two-dimension space, there are positions of initial nodes (red asterisk) and their moving targets (blue square). The proposed DPSO generates the moving plan and connects each starting position with its target, to reach a shorter moving distance for the whole network.

Unlike the binary PSO (Del Valle et al. 2008) where each particle takes YES/No decisions and the candidate solution is a combination strategy, DPSO in this paper aims at the problem where the solution is a sequence, i.e., an arrangement or a permutation. In detail, when solving the moving planning problem for the WSN layout, the initial positions of the nodes are treated as a fixed sequence and the solution is another arranged sequence where the corresponding positions of each dimensions in these sequences are "paired".

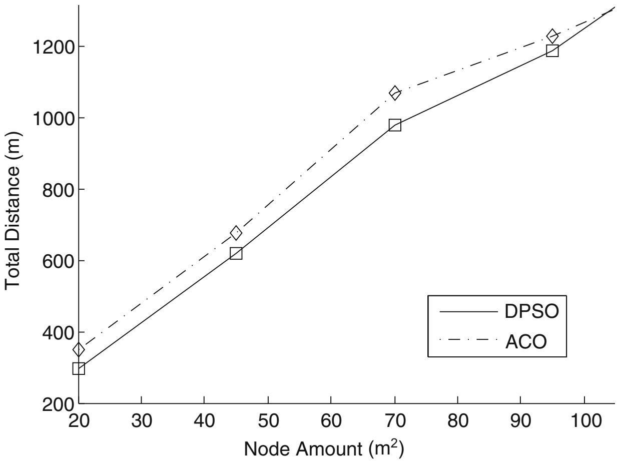

Fig. 2 Performance comparison of DPSO and ACO at different node amount

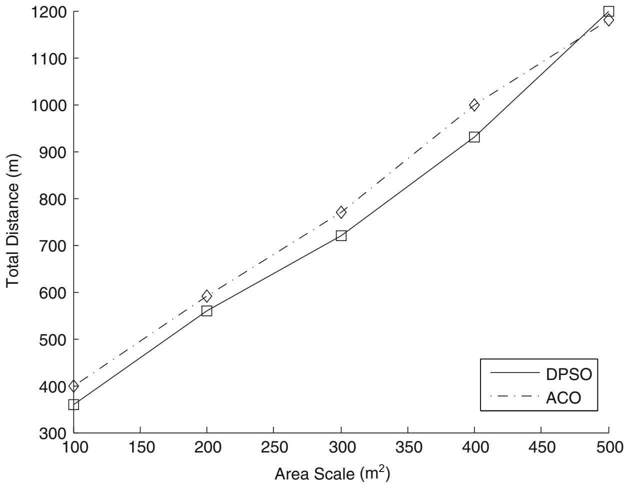

Fig. 3 Performance comparison of DPSO and ACO at different area scope

Compared to the previous DPSO in literature (Wang et al. 2003; Shi et al. 2007), this kind of DPSO does not introduce any external method to enhance the performance and just redefines the operators based on traditional PSO, which could be easier to implement. The proposed DPSO in this paper is a probabilistic-based method (means the result of an operation can be varying), this could lead to diversity and an escape from local optimum that fixed operations may cause. Figures 2 and 3, which are detailedly stated in the previous work (Du et al. 2014), shows the performance curve of this DPSO compared to another excellent discrete solver, ant colony optimization (ACO) with different problem scale. It can be seen that during a certain scale (when the node amount is under 100 and the ROI scale is less than 450 in this experiment), DPSO proposed in this paper achieves a shorter distance for moving planning, which means a better result.

Parameters in Eqs. (13), (16) and (17) are set as follows: $\omega = 0.6, c_1 = 2.2, c_2 = 1.4, \alpha = 0.6, \beta = 0.4, \gamma = 0.6, \alpha' = 0.7, \beta' = 0.7, \gamma' = 0.6$. It should be noticed that adjustment to this parameters may be needed in different situations, influenced by the sensing ranges of the nodes and the shape of ROI etc. In such cases the algorithm outcomes may also be different.

Since this kind of algorithm is easy to handle and performs well, it is used as a standard computation method to calculate the moving distance when evaluating a solution.

3.3.2 Heterogeneous multi-swarm PSO

This paper introduced a heterogeneous multi-swarm method to balance the exploration and exploitation of the algorithm. In the proposed algorithm, the population are divided into three sub-swarms with different evolutionary strategies, which include PSO with inertia weight, PSO with constriction factor and dynamic probabilistic PSO. Since every sub-swarm adopts a unique evolutionary strategy which has its own advantage, the diversity of the whole population is guaranteed. During the evolution, communication among the sub-swarms is maintained to avoid trapping into a local optimum.

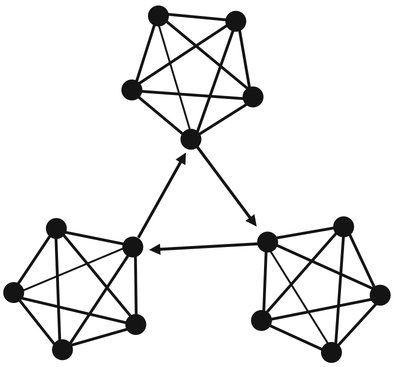

Fig. 4 Topology structure schematic diagram of heterogeneous multi-swarm strategy

Figure 4 shows the topology structure of the whole population. It is a fully-connected structure in each sub-swarm. And communication mechanism is maintained among the sub-swarms. In the generations with a certain interval (named alternate generation), sub-swarms will alternate their optimal solutions. Similar to the multi-swarm method in literature (Vanneschi et al. 2010), the best particles from a swarm replace the positions of the worst particles in the target swarm, and the particle amount in each swarm is constant. On the other hand, the alternate sequence of the swarms is random, which means one swarm has equivalent probability to attain the optima from every other swarms. In this way each swarm can absorb a different evolutionary direction guided by another swarm with different evolutionary strategies.

Algorithm 1 Process of heterogeneous multi-swarm PSO

```
Divide the whole population into three sub-swarms $S_1, S_2, S_3$
Initialize the positions of the population and update the gbest position
while $t < maxiter$ do
    for each $sub - swarm\ \ m \in population$ do
        for each $particle\ \ i \in sub - swarm\ \ m$ do
            Update the particle's position with the corresponding PSO algorithm
            Calculate the coverage rate $C$ with the coverage model(binary model or probabilistic model)
            Calculate the total moving distance $d$ using the DPSO algorithm
            Evaluate the fitness value with the fitness function (8)
            Update the personal best value ($pbest$) of the particle
        end for
        Update the sub-swarm best value ($ssbest_m$) of the sub-swarm
    end for
    if An alternate generation is met then
        for each $sub - swarm\ \ m \in population$ do
            $ssbest_m \leftarrow ssbest_{(m+1)\%3}$
        end for
    end if
    Update the global best ($gbest$) information of the population
    if a termination condition is met then
        Break the loop
    end if
end while
return $gbest$
```

Every particle in the population stands for a set of positions in a two-dimension space. For a node K, its position coordinate is presented as (x_k, y_k), thus for N nodes, a $2N$-dimensional particle $X_i = \{x_1, y_1, x_2, y_2 \ldots x_N, y_N\}$ stands for their positions. Algorithm 1 states how the heterogeneous multi-swarm PSO algorithm is processed.

3.3.3 *Analysis of the proposed algorithms*

Though a heterogeneous multi-swarm strategy is used, time complexity of this algorithm is equal to the traditional method. Assume the complexity of DPSO proposed in Sect. 3.3.1 is $\Theta(D)$ where D is a polynomial of problem space (the amount of nodes) n, the complexity of the whole MPSO can be expressed as $\Theta(t \cdot (mnD + k))$ where t is the maximum iteration times, m stands for the swarm size, and k is a leaner complexity expression standing for the consumption of information exchanging among the sub-swarms. With the same time consumption, a better diversity is achieved by the heterogeneous structure.

At the same time, the reason to combine those three kinds of algorithm should also be emphasized. In fact, PSO with inertia weight and constriction factor are two “basic” versions of PSO. The inertia weight decides whether the swarm should focus on exploitation or exploration, and the constriction factor also helps the convergence of the process. But DPPSO is quite different because it reconstructs the update method and hybrids the searching procedure with probabilistic distribution. This could enhance the searching ability in a new way. Collaboration of these three methods brings the absolute diversity to the whole population and swarms evolve cooperatively.

4 Experiments and analysis

4.1 Experiment settings

Simulation environment is built in this paper on two-dimension non-obstacle space. This paper compared the performance of three PSO algorithms which include traditional PSO, co-evolutionary particle swarm optimization (briefed as CPSO) (Kou et al. 2009) and the heterogeneous multi-swarm PSO that is introduced in this paper aimed at deployment problem.

Initially, 60 stationary nodes are randomly deployed in the WSN and there are 40 mobile nodes, with the scale width $S = 100$ m. The binary detection model and the probabilistic detection model are both taken into consideration. For binary model, sensing radius of a sensor $R_s = 7$ m, with $R_e = 0$ m. For probabilistic model, $R_s = 7$ m, and the detecting uncertainty radius $R_e = R_s/2 = 3.5$ m. $\alpha_1 = 1, \alpha_2 = 0, \beta_1 = 1, \beta_2 = 0.5, c_{th} = 0.9$. 1200 covering points are chosen randomly in the region which are to be covered. For these three PSO algorithms, they all contain 200 particles and the number of evolutionary iterations is 200.

4.2 Results and analysis

Figures 5 and 6 respectively illustrate the change of total moving distance of the network and its coverage rate of three PSO algorithms in binary model and probabilistic model. According to the presumption, a higher coverage rate and a shorter moving distance demonstrate a better performance, which is positively correlated to the *QoS* of the network. As can be seen in the figures, gaps exist

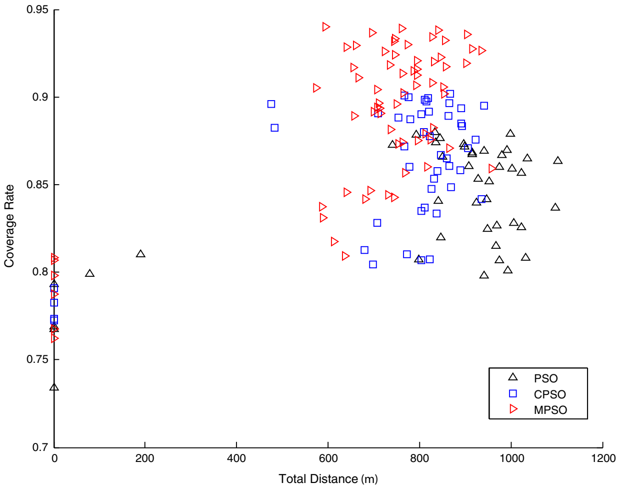

Fig. 5 Distance-coverage rate diagram in the binary model. (Color figure online)

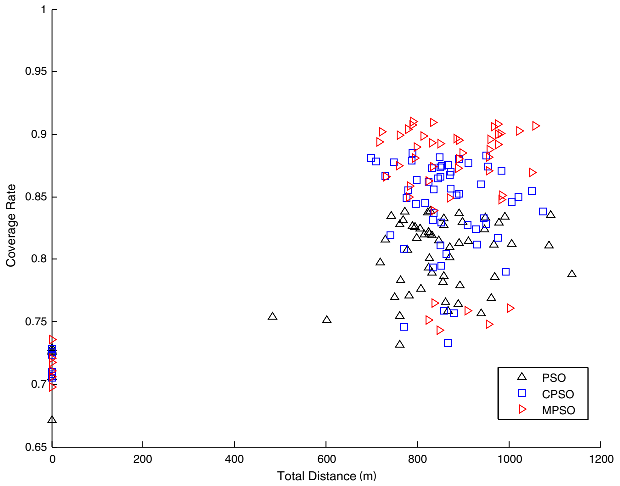

Fig. 6 Distance-coverage rate diagram in the probabilistic model. (Color figure online)

among the performances of those PSO algorithms. MPSO (red triangles in the figures) plays the best role, whose distribution is on a more upper left position, CPSO (blue squares in the figures) acts in the middle, while the traditional PSO (black triangles in the figures) still needs to be improved.

The fitness function shown in Eq. (8) magnifies the difference of the algorithms. Figures 7 and 8 demonstrate the fitness curves which grow with the iterations, and they are coincident with the distance–coverage rate figures. It can be found that PSO and CPSO converge in an earlier stage compared to MPSO. For MPSO, it has a fast convergence speed in the early stage and still has abilities to develop in the latter iterations (over 80), which is the consequence of maintaining the population diversity. The statistics of coverage result are given in Tables 1 and 2.

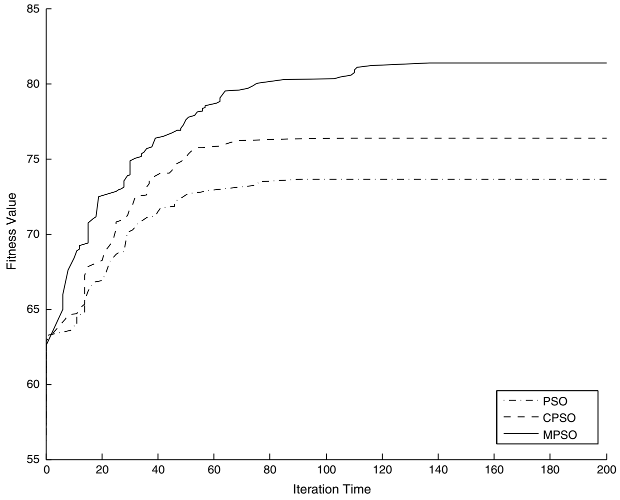

Fig. 7 Comparison of fitness value in the binary model

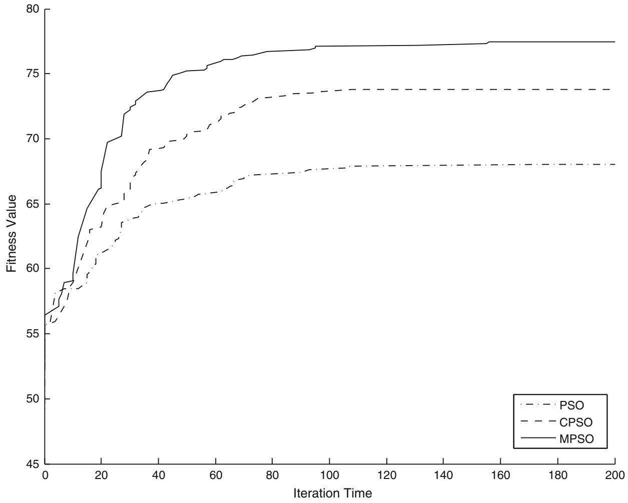

Fig. 8 Comparison of fitness value in the probabilistic model

Table 1 Performance comparison in binary model

Algorithm	Coverage rate (%)	Fitness value
MPSO	94	77.44
CPSO	90	74.26
PSO	89	68.55

Table 2 Performance comparison in probabilistic model

Algorithm	Coverage rate (%)	Fitness value
MPSO	91	77.47
CPSO	88	76.36
PSO	84	73.62

Under a sole PSO algorithm, the particle swarm can be easily trapped into local optimum, while the sub-swarms exchange different information, which can help the population keep away from the local optimal and lead to a broader exploration.

Experiment results demonstrate the importance of population diversity in the evolutionary algorithms. Information from different evolutionary strategies can change the convergence direction of the population and even help avoid trapping into local optimum. In the initial stage, different evolution strategies produce a faster search, which result in the high-speed rise of the fitness value, and the continuous impact among the populations contributes to a slower convergence in the late stage which leads to a higher ending result.

5 Conclusion

In the research area of WSN dynamic deployment, the moving distance of mobile nodes is often overlooked. This paper adds consideration of the moving distance, and provides a new way to solve the dynamic deployment problem. Specifically, a multi-swarm PSO is constructed by different PSO variants with different features, and different sub-swarms coevolve in the evolutionary process. Experiments show that, although the proposed method requires a little longer convergence time, the obtained solution is more satisfying. It also can be seen that collaboration between sub-swarms with different variants is a good strategy to upgrade the solving performance, and the proposed strategy is a good and feasible solution in the application of dynamic deployment.

For the node deployment in WSN, the uniformity of nodes and the topology design of communication should not be ignored, and it is the focus of our future work to consider. For the multi-swarm PSO to the dynamic deployment problem, more combinations of PSO variants and communication strategies among sub-swarms will be designed and attempted to improve the algorithm's performance in the area of node deployment.

Acknowledgments This paper is supported by National Natural Science Foundation of China (Grant No. 61170164).

References

Aleksandra M, Gavrilovska L (2011) WSN coverage and connectivity improvement utilizing sensors mobility. In: Proceedings of the 11th European wireless conference on sustainable wireless technologies

Aziz N, Mohemmed AW, Sagar BSD (2007) Particle swarm optimization and voronoi diagram for wireless sensor networks coverage optimization. In: Proceedings of the international conference on intelligent and advanced systems, pp 961–965

Blackwell T, Branke J (2004) Multi-swarm optimization in dynamic environments. In: Applications of evolutionary computing, Springer, Berlin, pp 489–500

Clerc M (2004) Discrete particle swarm optimization, illustrated by the traveling salesman problem. In: New optimization techniques in engineering. Springer, Berlin, pp 219–239

Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73

Del Valle Y, Venayagamoorthy GK, Mohagheghi S, Hernandez J-C, Harley RG (2008) Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans Evol Comput 12(2):171–195

Du HM, Ni QJ, Pan QQ, Yao YY, Lv Q (2014) An improved particle swarm optimization-based coverage control method for wireless sensor network. In: Proceedings of the 5th international conference on swarm intelligence, Springer, Berlin, pp 114–124

Kennedy J (2005) Dynamic-probabilistic particle swarms. In: Proceedings of the 7th annual conference on genetic and evolutionary computation (GECCO) ACM, pp 201–207

Kou XL, Liu SY, Zhang JK, Zheng W (2009) Co-evolutionary particle swarm optimization to solve constrained optimization problems. Comput Math Appl 57(11):1776–1784

Kulkarni RV, Venayagamoorthy GK (2011) Particle swarm optimization in wireless-sensor networks: a brief survey. IEEE Trans Syst Man Cybern Part C Appl Rev 41(2):262–267

Li SJ, Xu CF, Pan WK, Pan YH (2005) Sensor deployment optimization for detecting maneuvering targets. In: Proceedings of the 8th international conference on information fusion, vol 2

Liang JJ, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer. In: Proceedings of IEEE swarm intelligence symposium (SIS), pp 124–129

Liu YN, Wang G, Chen HL, Dong H, Zhu XD, Wang SJ (2011) An improved particle swarm optimization for feature selection. J Bionic Eng 8(2):191–200

Ma M, Yang YY (2007) Adaptive triangular deployment algorithm for unattended mobile sensor networks. IEEE Trans Comput 56(7):946–847

Mukhopadhyay S, Banerjee S (2012) Global optimization of an optical chaotic system by chaotic multi swarm particle swarm optimization. Expert Syst Appl 39(1):917–924

Ni QJ, Deng JM (2011) Two improvement strategies for logistic dynamic particle swarm optimization. In: Proceedings of the 10th international conference on adaptive and natural computing algorithms. Springer, Berlin, pp 320–329

Shi YH, Eberhart R (1998) A modified particle swarm optimizer. In: IEEE world congress on computational intelligence (WCCI), pp 69–73

Shi XH, Liang YC, Lee HP, Lu C, Wang QX (2007) Particle swarm optimization-based algorithms for TSP and generalized TSP. Inf Process Lett 103(5):169–176

Solomon S, Thulasiraman P, Thulasiram R (2011) Collaborative multi-swarm pso for task matching using graphics processing units. In: Proceedings of the 13th annual conference on genetic and evolutionary computation (GECCO), pp 1563–1570

Vanneschi L, Codecasa D, Mauri G (2010) An empirical comparison of parallel and distributed particle swarm optimization methods. In: Proceedings of the 12th annual conference on genetic and evolutionary computation (GECCO), ACM, pp 15–22

Wang K-P, Huang L, Zhou C-G, Pang W (2003) Particle swarm optimization for traveling salesman problem. In: Proceedings of the international conference on machine learning and cybernetics, vol 3, pp 1583–1585

Wang X, Wang S, Ma JJ (2007) An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. Sensors 7(3):354–370

Yazdani D, Nasiri B, Sepas-Moghaddam A, Meybodi MR (2013) A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. Appl Soft Comput 13(4):2144–2158

Zhao SZ, Suganthan PN, Pan Q-K, Tasgetiren MF (2011) Dynamic multi-swarm particle swarm optimizer with harmony search. Expert Syst Appl 38(4):3735–3742

Zou Y, Chakrabarty K (2003) Sensor deployment and target localization based on virtual forces. In: Proceedings of the 22nd annual joint conference of the IEEE computer and communications (INFOCOM), vol 2, pp 1293–1303