

# VD-PSO: An efficient mobile sink routing algorithm in wireless sensor networks

Wei Wang<sup>1,2</sup> · Haoshan Shi<sup>1</sup> · Dajun Wu<sup>3</sup> · Pengyu Huang<sup>4</sup> · Baojian Gao<sup>2</sup> · Fuping Wu<sup>5</sup> · Dan Xu<sup>2</sup> · Xiaojiang Chen<sup>2</sup>

Received: 16 May 2016 / Accepted: 6 September 2016 / Published online: 6 October 2016  
© Springer Science+Business Media New York 2016

**Abstract** Efficient energy consumption is crucial for energy constrained networks such as Wireless Sensor Networks (WSN). Using a mobile sink to collect the data of the nodes is a good method to balance the energy level of the nodes and prolong the lifetime of the whole network. For the mobile sink, an efficient path planning can make the mobile sink visit significantly more nodes during a limited period and shorten the latency of information gathering. Considering the communication range of the nodes, we can deduce this routing problem as a special case of traveling salesman problem with neighborhoods (TSPN), which is a NP-hard problem [1]. In this paper, we propose a novel routing design algorithm based on Variable Dimension Particle Swarm Optimization (VD-PSO). In this algorithm, every feasible path solution of TSPN is expressed as a particle. Each dimension of the particle is the coordinates of a rendezvous point (RP, the point where the mobile sink stays to gather data). The dimensionality of the particle is equal to the number of the rendezvous

points in the path. Using the evolutionary method of the particles, we can derive the optimal path of the mobile sink. Simulation results show that the proposed algorithm has fast convergence speed, and the result is quite approximate to the optimal solution.

**Keywords** Wireless sensor networks · Mobile sink · Traveling salesman problem with neighborhoods · Variable dimension particle swarm optimization

## 1 Introduction

In recent years, wireless sensor networks (WSN) have been widely used and deployed. The sensor nodes in WSN function to sense and gather the information of the field and are utilized in scenarios such as environment monitoring, battlefield surveillance, intrusion detection, etc. [2–4]. According to

---

✉ Pengyu Huang  
pyhuang@mail.xidian.edu.cn

Wei Wang  
wwang@nwu.edu.cn

Haoshan Shi  
shilaoshi@nwpu.edu.cn

Dajun Wu  
djwu@i2r.a-star.edu.sg

Baojian Gao  
gaobj@nwu.edu.cn

Fuping Wu  
fpwu@xidian.edu.cn

Dan Xu  
xudan@nwu.edu.cn

Xiaojiang Chen  
xjchen@nwu.edu.cn

<sup>1</sup> School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China

<sup>2</sup> School of Information and Technology, Northwest University, Xi'an, China

<sup>3</sup> Agency for Science, Technology and Research, Institute for Infocomm Research, Singapore, Singapore

<sup>4</sup> School of Telecommunication and Engineering, Xidian University, Xi'an, China

<sup>5</sup> School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, China

Gartner, a US market-research firm, there will be approximately 26 billion devices on the Internet of Things (IoT) by the year 2020 [5]. As the main component of the IoT, WSN could bring more than 30 billion-dollar revenue. Therefore, large amount of funds have been invested in WSN research and development by governments, NGOs and many MNCs such as Google, Cisco, Apple, etc. Up to present, there have been many successful cases. For example, Liu et al. have deployed more than 330 nodes in the wild for forest surveillance [6] and the system was dubbed as GreenOrbs. Similarly, more than 42 Tiger reserves based on WSN have also been established in India [2].

With the rapid development of unmanned vehicles and unmanned aircrafts, sink nodes can be deployed on these unmanned mobile devices and act as mobile sinks. These mobile sinks extend the mobile and dynamic property of WSNs [7]. In traditional WSNs with static sinks, sensors are equipped with batteries and forward data through multi-hops to the sink [8, 9]. Thus, the nodes close to the sink will deplete their batteries faster than the other nodes due to higher data relaying load [10]. As a result, this will lead to unexpected network disconnection and shortening of network lifetime [11]. But with the introduction of mobile sink, data can be collected in a single-hop manner while mobile sink moves through a sensing field. Thus many studies have shown that mobile sink can balance the traffic load, prolong the network's lifetime, and improve connectivity and positioning capabilities as well [12, 13].

However, the benefits stated above are mainly based on the premise that a comprehensive and effective solution exists for the routing design of the mobile sink [14]. Efficient mobile sink routing protocol should be able to find the shortest path. The reason is that the energy of the mobile sink is limited and a short path can reduce energy consumption of the mobile sink. On the other hand, a short path can also make the mobile sink collect all the data in a short time. In a word, minimization of latency is of great importance, especially in real-time applications.

In this paper, we consider a WSN with multiple static sensor nodes and one mobile sink, and subsequently research how to find the shortest path to the mobile sink. This problem can be formulated into a travelling salesman problem (TSP) [15] when the communication range of the nodes are neglected. In such case, we have to seek the shortest path that traverses each sensor node. The mobile sink can visit every node along this path and at the same time collect the monitor data. Quite a number of relevant approaches have been proposed in the literature. Gupta et al. and Xing et al. [16, 17] studied on a rendezvous-based data collection approach which takes a subset of sensor-nodes as the rendezvous points where the mobile sink stays and gathers the data from the sensors. When mobile base station arrives at the rendezvous point, the

information of the sensor node will be transmitted to the mobile base station. Similarly, Liu et al. [18] searches the rendezvous points within a hop bound.

Considering the communication range of the mobile sink and the sensor nodes, the problem of the best path planning can be transformed to a special case of travelling salesman problem with neighborhood (TSPN) [19, 20]. In this case, the mobile sink does not need to visit the location of every sensor node on its path. It just needs to pass the point which is involved in communication range of one or more sensor nodes. As demonstrated in [21], the path designed to consider the communication radius will be more flexible and the length of the path can be reduced, while greatly improving the performance of the network.

In this paper, we introduce a novel routing algorithm for mobile sink. The algorithm fully considers the communication ranges of sensor nodes and the mobile sink, and functions based on VD-PSO. In this algorithm, the coordinates of each RP on the path of the mobile sink are treated as a dimension of one particle. Thus the dimensionality of this particle will be determined by the number of RPs on the path. We can see that each particle is a solution, and a swarm of particles formulate a solution space. With the evolution proceeding, we are able to obtain the optimal solution which is the shortest path of the mobile sink.

To the best of our knowledge, people rarely consider how to find the shortest path in the non-uniformly distributed WSN. However in practice, we have observed that the distribution of sensor nodes is always uneven, especially in wide use. E.g. in military surveillance and environmental monitoring [22], random throwing of sensor nodes is widely applied. As in protection of wall painting in Italy, the locations of the sensor nodes are all dependent on the location of the paint [23]. In these cases, it is hard to control the topology of the network and make it homogenized. Moreover, this non-uniform distribution will make the number of RPs in each path quite different, that is to say the dimensionality of each particle is different. This will make it impossible to express the particle as before, and the traditional iterative formula is also unsuitable for our algorithm. In this scenario, our approach applies variable dimension (VD) to each particle, and adjusts the iterative formula to update it.

During the running time, we discover that the dimensionality of one particle could become closer. That is to say there is more RPs in small areas. Since this is detrimental to energy efficiency, we set a RP merging strategy to deal with this situation. In addition, we also use reverse order and simulated annealing algorithm to improve the convergence speed and precision of the solution. Our simulation results show that VD-PSO can find the shortest path effectively and perform efficiently, especially in non-uniform network deployment.

The main contributions of this paper are summarized as follows:

- We design a routing algorithm for the mobile sink based on variable dimension PSO. It describes each possible path as a particle with variable dimension and finally finds out the shortest path for the mobile sink.
- We propose a special strategy to deal with the update proceeding of the particles with different dimensionality. This method makes our algorithm work well in non-uniform network.
- We further improve the performance of the algorithm by incorporating reverse order and simulated annealing method.

The remainder of this paper is organized as follows. Section 2 formulates the problem we aim to resolve. Section 3 divides the main problem into some subproblems, describes key steps of our proposed algorithm for the problem and illustrates on how to optimize it. Simulation procedure and result analysis are provided in Section 4. Finally, Section 5 concludes the whole paper.

## 2 Problem formulation

As mentioned previously, the energy of the mobile sink is limited. Thus, if the transceiver circuit of the mobile sink keeps working all the time during the movement, its battery power will run out quickly. Therefore, in practice, we often put the transceiver in sleep mode and wake it up only when the mobile sink arrives at a RP on its path. In this way, the problem of finding the best path for the mobile sink can be divided and formulated as two subproblems:

- (1) How to choose the location of the RP?
- (2) How to decide the shortest path which passes through each RP?

According to [21], we know that the first one is a NP-hard problem, and the second subproblem is NP-hard as well. In other words, finding the best travel path of the mobile sink in a large scale WSN is extremely difficult, and it has no polynomial-time solution.

### A. Finding the best traveling path of the mobile sink

We consider one WSN composed of  $N$  static sensor nodes. These  $N$  nodes are deployed randomly in an area of  $R_X \times R_Y$ .  $R_X$  and  $R_Y$  are the ranges of horizontal and vertical coordinates of the deployment area respectively. The communication range of the mobile sink and sensor nodes are all  $R$ .

We set  $T_v$  as one of the paths that the mobile sink travels through. The path consists of multiple RPs and the lines between them. As a result, we can formulate the path as an ordered set of RPs. We denote

$$v_i = \{(x_i, y_i) | 0 < x_i < R_X, 0 < y_i < R_Y\}$$

as the coordinate of RP  $i$ . A path of mobile sink to move can be characterized by a unique ordered set  $T_v: \langle v_1, v_2, \dots, v_n \rangle$ , where  $|T_v| = n$  is the number of RPs on the path. And  $d_{ij}$  is the distance between the RP  $i$  and  $j$ .

$$d_{iX} = \min_{k \in X, k \neq i} (d_{ik}) \quad (1)$$

$d_{iX}$  is the shortest distance between node  $i$  and point set  $X$ .  $d(T_v)$  is the length of the path.

We can define the task of finding the best path as:

- Find the shortest path  $T_{opt}$  of which the starting-point is  $S$  and the traversing finally returns to  $S$ .
- The mobile sink must pass through the coverage of each node at least once, that is, for each sensor node  $i$  there should be  $d_{iT_{opt}} \leq R$ .

### B. Traditional PSO

Particle Swarm Optimization (PSO) is a kind of optimization algorithm based on swarm intelligence. It was proposed in 1995 by Kennedy and Eberhart, who were American psychologist and electrical engineer respectively [24]. In traditional PSO, each particle keeps on learning from past experience of itself and other member's experience to approach the optimal solution.

During running time, each particle will trace two optimal solutions at the same time. One is  $pbest$ , the best solution that each particle has found until now, and the second is  $gbest$ , the best solution that all particles have obtained currently. We assume that the location of  $i$ -th particle in  $n$ -dimensional space is  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ . The speed of  $i$ -th particle is  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ , and the best location it has got is  $pbest_i = (p_{i1}, p_{i2}, \dots, p_{in})$ . The best location of the group is  $gbest_i = (p_{g1}, p_{g2}, \dots, p_{gn})$ .

Each particle updates its speed and location according to formula (2) and (3).

$$v_{id}^{k+1} = \omega \times v_{id}^k + c_1 \times r_1 \times (pbest_{id}^k - x_{id}^k) + c_2 \times r_2 \times (gbest_{id}^k - x_{id}^k) \quad (2)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^k \quad (3)$$

$x_{id}^k$ : The location of  $d$ -th dimension of particle  $i$  at time  $k$ .

$v_{id}^k$ : The speed of  $d$ -th dimension of particle  $i$  at time  $k$ .

$\omega$ : Inertia weight.

$c_1$ : The step-length for the particle which follows to the best location of itself.

$c_2$ : The step-length that the particle which follows to catch the best location of the group.

$r_1$  and  $r_2$ : Random values between  $[0,1]$ .

In addition, we set speed interval as  $[v_{\min}, v_{\max}]$  and location interval  $[x_{\min}, x_{\max}]$  to limit the movement of the particle.

As one of the excellent optimization algorithms, PSO has features of both simplicity and fast convergence. However, this traditional PSO cannot be used directly to solve the routing problem of the mobile sink based on our subsequent analysis.

### 3 Finding the best path with VD-PSO

As analyzed earlier, our goal is to find a traveling path for the mobile sink, and the mobile sink will visit each sensor node at the RPs on the path. In order to solve this problem by PSO, the key question lies in how to establish the relationship between the particle and the traveling path. In other words, a possible path should be represented as a particle that is one feasible solution. Our proposed approach uses a variable dimension particle to express one possible path, and the location of every RP on the path becomes one dimension of the particle. Since every particle may have different dimensionality, there is no way to know in advance how many RPs exist on the path. This results in the fact that the update formula (2) and (3) cannot be applied directly.

In order to solve this routing problem based on PSO, it is required to address the following questions:

- Since we do not know how many RPs should be used, we cannot use the traditional PSO with fixed dimensionality. Thus the question lies in how to represent a feasible solution using PSO method in our problem.
- In a large scale WSN, how to generate an initial particle reasonably and effectively is an important issue.
- Because the dimensionality of each solution may be different, we must find out how to update the speed and location of each particle.
- As the number and the location of RPs change, we need to search in multidimensional solution space of PSO. In this condition, what can be done to accelerate the algorithm and perform more effectively?

The following sections will provide solutions for these problems one by one.

#### A. Solution Presentation and Update Procedure

We can regard one of the traveling paths as a sequence of connected line segments and use  $v_i = \{(v_{ix}, v_{iy}) | 0 < v_{ix} < R_X, 0$

$< v_{iy} < R_Y\}$  as the coordinates of the RP  $i$  on the path. We can then indicate the  $k$ -th travelling path by an ordered set of location coordinates  $T_{(k,n_k)} : \{v_{k1}, v_{k2}, \dots, v_{kn_k}\}$ , where  $n_k$  is the number of RPs on path  $k$ . Because for each particle the  $n_k$  may be different, as compared with traditional PSO, the particle here has variable dimensionality, and the number of RPs equals the dimensionality of the solution.

As discussed previously, during running time of PSO, each solution will track the best position of each particle ( $pbest$ ) and the best position of the group ( $gbest$ ).  $n_k$  in each particle is different, and thus we cannot use the traditional formula to update the speed and location of the particle. Herein, we propose a novel method that enables the particle to trace the  $pbest$  and  $gbest$  efficiently.

When one dimension of the particle needs to update its speed and location, the following can be performed: at first, search every dimension and find the nearest dimension (position) in  $pbest$  and  $gbest$ . We can subsequently update the dimension of the particle based on these three position vectors. The detailed flow is given below.

We assume that the dimension  $j$  of particle  $m: T_{(m,n_a)}$  needs to update the speed and location, and the best solution of its own is  $pbest_{(m,n_b)}$  until now. Now we search every dimension of  $pbest_{(m,n_b)}$  to find the dimension which is nearest to dimension  $j$  and record it as  $k$ . At the same time if the best solution of the group is  $gbest_{(n_c)}$ , we search it to find the dimension which is nearest to dimension  $j$  and record it as  $l$ . Thereafter, we update the speed and location of particle  $m$  according to the formula (4) and (5).

$$v_{mj}^{j+1} = \omega \times v_{mj}^j + c_1 \times r_1 \times (pbest_{mk}^j - x_{mj}^j) + c_2 \times r_2 \times (gbest_l^j - x_{mj}^j) \quad (4)$$

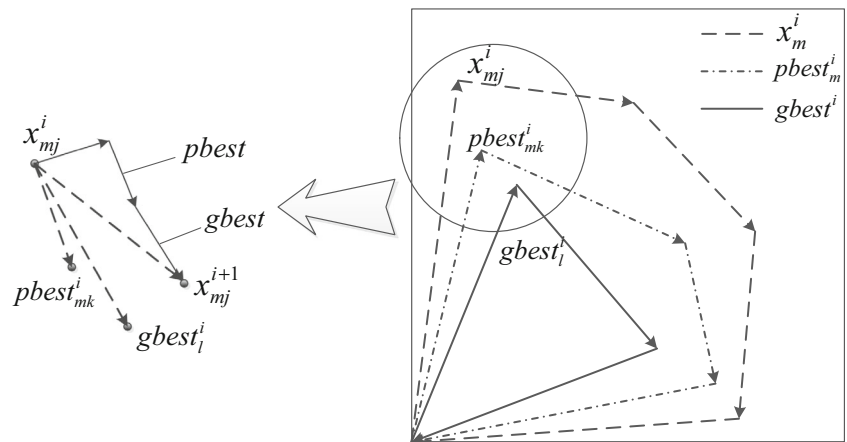
$$x_{mj}^{j+1} = x_{mj}^j + v_{mj}^j \quad (5)$$

The trace method is illustrated in Fig. 1.

#### B. Algorithm acceleration

During the running time, each particle will search in the multi-dimension solution space using the update method described above. Within the updating process of the particle, the spatial position of each particle will approximate to the spatial position of the global optimum particle step by step. As a result, we notice that some of the dimensions of one particle will get increasingly closer, which means the optimal solution of TSPN tends to have less dimensionality. Moreover, the less dimensionality of the particle has, the shorter length of the

**Fig. 1** Updating process of tracing the nearest dimension



path will be. Nevertheless, in this process, as some dimensions get closer, it also means that the RPs on the path will be closer. This would lead to the result of which several RPs are located in a very small area. Our simulation results also show that this phenomenon would happen, and in some cases, it occurs more frequently.

On the other hand, as mentioned, in a real application scenario, while the mobile sink collects the packets along the route, the transceiver of it does not work all the time. Generally, it only wakes up and starts to work at the moment when the mobile sink arrives at each RP. When the mobile sink leaves the RP, the transceiver will be put into sleep mode or closed in order to save energy. Hence, if the RPs get too close, the mobile sink would wake up and close the transceiver frequently. This is not a good solution with regard to energy-saving and system efficiency.

In order to prevent this phenomenon from happening, we propose a strategy to merge adjacent RPs. In this strategy, we select the midpoint of segment constituting these two RPs as a potential RP that can be called *pRP*, if the distance between two RPs is less than a preset threshold. If over this potential RP the mobile sink's working region can cover the entire sensor nodes that original two RPs can cover, we replace the two adjacent RPs with the potential RP and delete these two RPs from route of the mobile sink. In order to introduce this strategy in our algorithm, we need to analyze this strategy initially and solve two key problems thereafter: The first is how close the adjacent RPs should be for them to be merged. The second is how to choose the replace point on the segment connecting two adjacent RPs.

Regarding the first problem, we find that when two adjacent RPs get close enough, there will be a large overlapping area in their working region. Hence if we select one point in the overlapping area of these two RPs as a new RP, the mobile sink's working area, at this point, would contain the sensor nodes including two original RPs with high probability. (Based on this idea and aiming to simplify the analysis, we

use one point on the segment that connects two adjacent RPs as the new RP to replace original two RPs.)

We first adopt the geometric method to analyze this problem. We use the segment's distance  $dist_{seg}$  as the standard of measurement. While  $dist_{seg}$  is less than a preset threshold  $dist_{THR}$ , we merge these two RPs. In order to analyze this problem efficiently, let  $v_i: (x_i, y_i)$  and  $v_j: (x_j, y_j)$  be the coordinates of two RPs which are close to each other. Let  $L$  be the length of the segment between  $v_i$  and  $v_j$ . Generally, we can assume that the working region of a mobile sink is the same, so it can be seen as a circle with radius  $R$ . We set the coverage area (working region) of RP  $v_i$  and  $v_j$  as  $C_i$  and  $C_j$ , similarly, set the overlap area as  $C_{ij}$ . In Fig. 2(a) to Fig. 2(c), it is obvious to find that  $C_{ij} = 0$  when  $dist_{seg} = 2R$ . And the formula  $C_{ij} = \pi R^2$  holds when  $dist_{seg} = 0$ . Whilst  $0 < dist_{seg} < 2\pi$ ,  $\pi R^2 > C_{ij} > 0$  holds.

Therefore, when  $C_{ij}$  is smaller, the probability of finding a suitable coverage region to contain the overlap region of two adjacent RPs is lower. According to the basic geometry theory, this overlapping region can be seen as a combination of two shapes of which each shape can be considered as a fan minus a triangle. Hence we can obtain

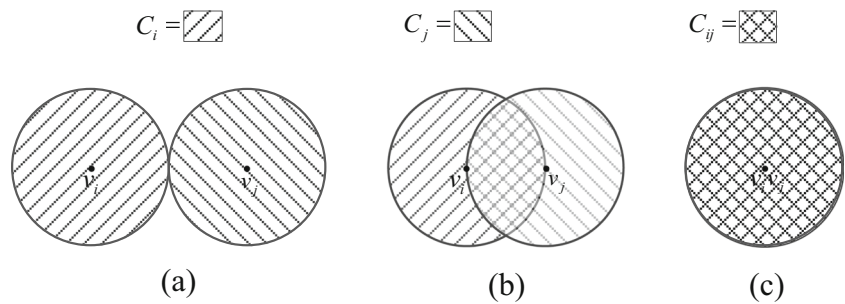
$$\alpha = 2\arccos\left(\frac{L/2}{R}\right)$$

$$C_{ij} = 2\left(\frac{\alpha}{360}\pi R^2 - \frac{L}{2}\sqrt{R^2 - \left(\frac{L}{2}\right)^2}\right)$$

In this formula,  $\alpha$  is the central angle degree corresponding to the fan. Based on this formula, we can illustrate the overlapping area vs.  $dist_{seg}$ , as shown in Fig. 3. From Fig. 2(b), it is evident that  $dist_{seg}$  is proportional to normalization of the overlapping area. Taking into consideration of algorithm complexity and success rate of merge strategy, we choose the  $dist_{seg}$  as the final  $dist_{THR}$  when  $C_{ij} = 0.8\pi R^2$  holds.

For the second problem, let  $v_k: (x_k, y_k)$  be the midpoint of segment that connects two RPs above and we still consider the

**Fig. 2** Overlapping area with different  $dist_{seg}$



working region of mobile sink as a circle with radius  $R$ . According to Fig. 4(a) to Fig. 4(c), it is evident that the circle does not cover all overlapping region if we choose an arbitrary point not on the segment between  $v_i$  and  $v_j$  as new RP. Consequently, we should choose a point on the segment connecting  $v_i$  and  $v_j$  as the new RP. However, the question now lies in how to determine which point is the best point. In fact, the node number covered by working region of  $v_i$  and  $v_j$  is usually not the same. Hence if the working region of  $v_i$  covers more nodes, it should select one point closer to  $v_i$  as the new RP and vice versa. Unfortunately, this step needs extensive computation. To simplify the algorithm implementation, we use the midpoint  $v_k$  of the line directly as the new RP.

C. Reverse order

In basic PSO Formula 2, in order to improve the precision of the algorithm, we often restrict the variation range of particle’s speed within a certain interval  $[v_{min}, v_{max}]$ . However at the same time, these restrictive measures also narrow the search scope of the particle. At the beginning of the operation, the initial path is generated randomly to increase the search scope. In a large scale WSN, almost every initial path is far away from the optimal path. If we restrict the speed of the particle, it will take long time to approach the optimal solution. On the other side, if we do not restrict the speed of the

particle, it is difficult for the final path to avoid being trapped into local optimum.

In this paper, we introduce reverse order strategy of parthenogenesis algorithm (IPGA) to improve searching speed of the particle without sacrificing the accuracy. In each iteration, each particle performs local tuning according to reverse order strategy. The results of simulation show that this strategy not only increases the searching speed of particles, but also reduces the time complexity of iterations.

In our algorithm, a particle is constituted of an ordered set of location coordinates. Reverse order strategy is equivalent to exchanging the order of some part of this set with a probability of  $P$ . Here the single-point reverse strategy is adopted. Initially, we generate random integer  $n_i$  and  $n_j$  for particle  $T_{(m,n_a)}$ , of which length is  $n_a$ . Thereafter, the order of the coordinates between  $n_i$  and  $n_j$  is reversed to generate a new particle  $T_{(m,n_a)}^*$ .

D. Simulated annealing algorithm

In traditional PSO algorithm, the particle converges fast at the beginning. However in subsequent iterations, convergence speed would slow down as the distance between particles becomes shorter. This would lead the particle to fall into a local optimal solution. In order to strengthen the search ability of particles, we introduce a simulated annealing method into the iterative process of VD-PSO. A bad solution (particle) will be accepted with a certain probability in each iteration.

After completing the previous steps of updating and merging, each particle evaluates the new and the old solutions and then it will accept a bad solution with a certain probability according to Formula (6).

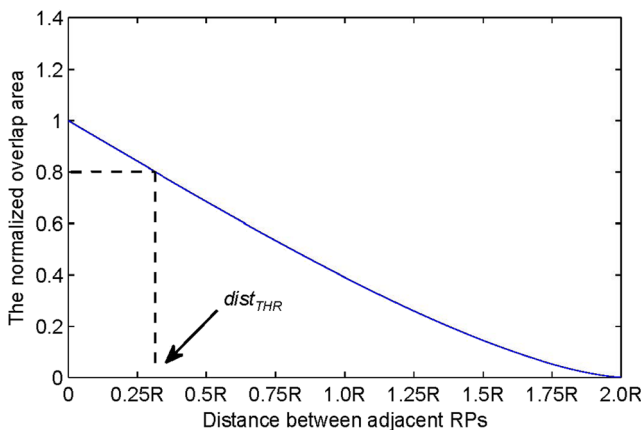
$$p = \begin{cases} 1 & C_1 \\ \exp\left(-\frac{E(T_{(m,n_m)new}) - E(T_{(m,n_m)old})}{T(t)}\right) & C_2 \end{cases} \quad (6)$$

$$T(t + 1) = k \times T(t) \quad (7)$$

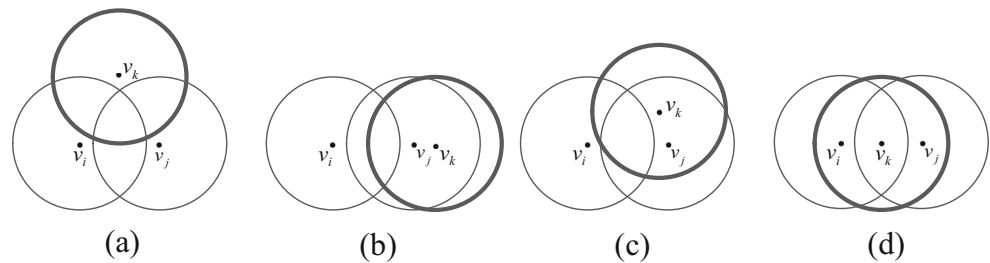
In Formula (6):

case C1: if  $E(T_{(m,n_m)new}) < E(T_{(m,n_m)old})$

case C2: if  $E(T_{(m,n_m)new}) \geq E(T_{(m,n_m)old})$



**Fig. 3** Selection of  $dist_{THR}$

**Fig. 4** Selection of new RP

$T_{(m,n_m)new}$  and  $T_{(m,n_m)old}$ : The particle before and after the previous steps of updating and merging.

$E(T_{(m,n_m)new})$  and  $E(T_{(m,n_m)old})$ : The fitness value of these two particles.

$p$ : The probability of accepting the new particle.

$T$ : The temperature of iteration.

Formula (6) shows that, if the fitness value of the new particle is lower than the old one, we replace the old particle by the new particle with probability 1. If the fitness value of the new particle is greater than the old one, we replace the old particle by the new particle with probability  $p$ .

Formula (7) shows the annealing temperature of the algorithm. At the beginning of iteration, we set a larger initial temperature  $T(t_0)$ , and  $k$  is positive and slightly less than 1. As the iteration continues, temperature  $T$  becomes lower gradually. Therefore the replacement probability also becomes lower when the new particle is worse than the old one. In the subsequent stages of iteration, the replacement probability approximates to zero since the temperature is very low.

#### E. Evaluation Function

The goal of this paper is to find out the best travelling path for the mobile sink. The length of the best path should be the shortest one. Therefore, we use the length of the path as the fitness value of the particle. The smaller the fitness value is, the better the particle would be and vice versa.

#### F. Algorithm Flow

We describe VD-PSO Algorithm for TSPN as follows:

Step1. Initialization of TSPN problem: Initialize the parameters of the TSPN problem, such as the sensor nodes number of the WSN, the size of simulation region, etc.

Step2. Initialization of VD-PSO algorithm: Set the number of particles, set the learning factors and a random velocity, then set generation  $t = 0$ . Each particle represents a random feasible solution.

Step3. Calculate each particle's fitness value and update the  $pbest$  and  $gbest$ .

Step4. For every dimension, each particle searches and finds the nearest dimension in its  $pbest$  and  $gbest$ , updates

the velocity and position according to Formula (4) and (5) based on these two nearest dimensions then generates one new particle.

Step5. Calculate the new and the old particles' fitness value for every particle. If the new particle's fitness value is less than the old one, use the new particle to replace the old particle.

Step6. Each particle searches every dimension of itself to find whether the distance between two adjacent dimensions is less than the threshold  $dist_{THR}$ . If so, combine these two dimensions using the adjacent RPs merge strategy and generate a new particle.

Step7. Calculate the new and old particles' fitness value again. If the new particle's fitness value is less than the old one, use the new particle to replace the old particle.

Step8. Each particle reverses some part of itself with some preset probability  $P$  and then generates one new particle.

Step9. Calculate the new and old particles' fitness value again. If the new particle's fitness value is less than that of the old one, use the new particle to replace the old particle.

Step10. Update each particle's fitness value and update the  $pbest$  and  $gbest$ .

Step11.  $t = t + 1$ , If  $t < M$ , go to step 4, and continue to iterate, or else, stop iterations. Now the  $gbest$  is the best solution which the algorithm has found.

## 4 Simulation

The simulation scene of the proposed approach is a square area of about 500 m on each side. The coordinate of the start point (SP) of the mobile sink is (250,250), that is, the start point is located at the center of the simulation scene. We assume that the working range of the mobile sink can be changed from 20 m to 100 m. There are 50 to 100 sensor nodes to be deployed randomly in the simulation region. We perform the simulation with different sensor node number for 50 times, and average all the results together.

In order to evaluate the performance of our algorithm, we mainly study the length of the path that the mobile sink travels.

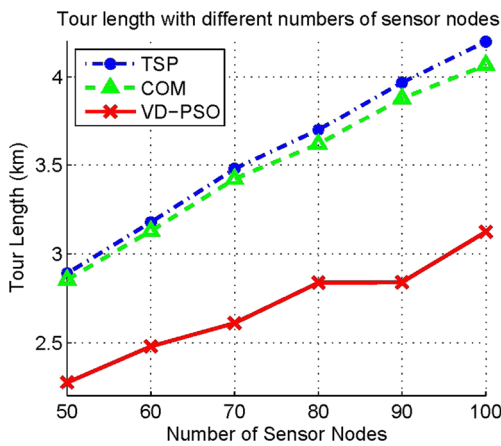


Fig. 5 Path length with different numbers of sensor nodes ( $R = 50$  m)

The path starts from the SP  $(x_j, y_j)$  and returns to this point after visiting each RP. As the path length becomes shorter, the mobile sink spends less time to gather data or recharge each node of the entire network. At the same time, the shorter path length means the lower energy consumption of the mobile sink and better maintainability of the whole system. Thus the length of the traveling path is crucial to evaluate the performance of this kind of TSPN algorithm.

In our simulation, we study the length of the path in two cases:

- (1) With the fixed working range of the mobile sink, we deploy different number of sensor nodes in the simulation field.
- (2) With the fixed number of sensor nodes, we change the working range of the mobile sink.

In order to evaluate the algorithm performance, we perform simulation using three different algorithms, TSP, COM (Combination Algorithm) and our proposed VD-PSO algorithm. TSP expresses the shortest path length that the mobile sink goes through each sensor node only once. COM

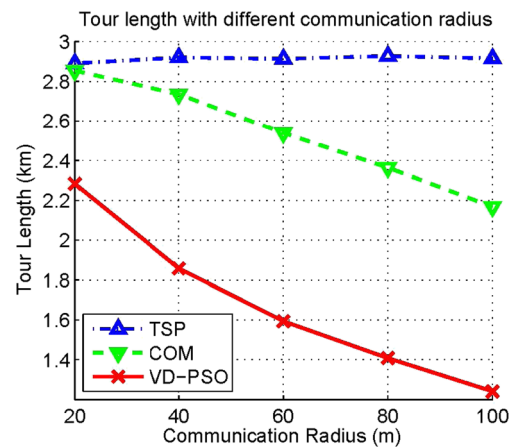


Fig. 7 Path length with different communication radius ( $N = 50$ )

algorithm originates from literature [12] and it combines the data collection while these sensor nodes can be covered by a RP’s communication range.

Initially, the working range of the mobile sink is set to 50 m and 100 m, 50 to 100 sensor nodes are deployed in the simulation field. In each case, we record the length of the path which is obtained by each algorithm.

As shown in Fig. 5 and Fig. 6, when the number of the sensor node increases, all path lengths obtained using these three algorithms increase. However, we can observe that the path length of TSP is the longest one among the three. This is due to fact that the TSP algorithm does not consider the working range of the mobile sink, and therefore the mobile sink must traverse the location of each sensor node. In other words, the location of each sensor node is exactly the same as the PR’s location in TSP algorithm. Thus as the number of the sensor nodes increases, the path length increases substantially.

The performance of COM is better than TSP algorithm. This is because COM algorithm can merge some PRs according to overlapping degree of the mobile sink’s working range, and COM algorithm can gradually reduce the number of RPs. Based on this, COM algorithm can reduce the path length to

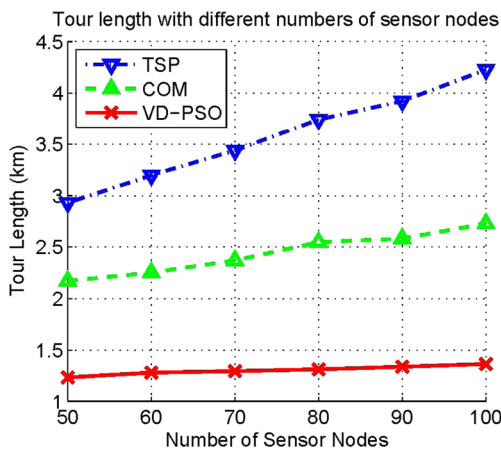


Fig. 6 Path length with different numbers of sensor nodes ( $R = 100$  m)

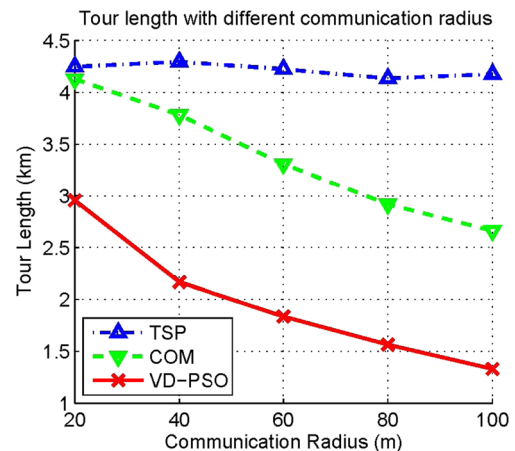


Fig. 8 Path length with different communication radius ( $N = 100$ )



some extent. Therefore the algorithm performance is improved partially.

According to Fig. 5, we can observe that the VD-PSO algorithm improves the performance significantly. When there are 100 sensor nodes in the network and the working radius is set to 50 m, the path length obtained by VD-PSO algorithm decreases nearly a quarter than COM. Figure 6 demonstrates the similar result, in which VD-PSO algorithm reduces the path length by about 45 %. In general, with the fixed working range of mobile sink, VD-PSO algorithm can greatly reduce the length of the path. This also means that VD-PSO algorithm can achieve much better performance.

In another setting, 50 and 100 sensor nodes are deployed in the same field, and the working range of the mobile sink is increased from 20 m to 100 m gradually. As illustrated in Fig. 7, the path length of TSP is the same as the value when working radius is fixed. For COM algorithm, it can merge more RPs as the working range of the mobile sink increases. Consequently, the path length decreases, since the working range is shortened. Based on the simulation configuration above, we can observe that the performance of VD-PSO is the best. It is able to achieve approximately 20 % performance improvement as compared with COM. Similarly, as shown in Fig. 8, VD-PSO algorithm can obtain 50 % performance improvement when compared with COM algorithm. This confirms the effectiveness and efficiency of our proposed algorithm.

## 5 Conclusion

In this paper, we propose a Variable Dimensions Particle Swarm Optimization algorithm to resolve the TSPN problem in WSN with a mobile sink. With this routing algorithm, the mobile sink could collect the data efficiently, and it will be helpful for us to locate the target [25, 26]. In the algorithm, we adopt a variable dimensions particle to represent the feasible solution of TSPN.

During the VD-PSO algorithm runtime, each particle needs to trace the best solutions of local and global optimal solutions which have different dimensions. Herein, we creatively adopt the nearest particle dimension fitting method to solve the updating problem and this significantly accelerates the searching speed of the particles. In the iteration process, the simulated annealing algorithm and reverse order method is introduced into VD-PSO algorithm. These two methods greatly improve the global searching ability of the VD-PSO algorithm and enable the particles to escape from the local extremum.

Simulation results demonstrate that the VD-PSO algorithm of this paper can solve the routing problem of the mobile sink in WSN effectively. The algorithm has the advantages of both fast convergence speed and good

stability. This study is significant for mobile sink routing design in large scale WSN, particularly the scenario of non-uniform deployment.

**Acknowledgments** This research was supported by the National Natural Science Foundation of China (61170218, 61501372, 61272120), the Education Department of Shaanxi Province Natural Science Foundation, China (15JK1742, 12JK0937), The Foundation of Northwest University (ND14041), Science Foundation of Shaanxi Xi'an Beilin (GX1403).

## References

1. Dumitrescu A, Mitchell JSB (2003) Approximation algorithms for tsp with neighborhoods in the plane. *Journal of Algorithms* 48(3): 135–159
2. R. Singh, G. M. Asutkar. (2015) Survey on various wireless sensor network techniques for monitoring activities of wild animals[C]. 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIECS). IEEE.
3. Zhang H, Cheng P, Shi L, Chen J (2016) Optimal dos attack scheduling in wireless networked control system. *IEEE Transactions on Control System Technology* 24(3):843–852. doi:10.1109/TCST.2015.2462741
4. Chen J, Xu W, He S, Sun Y, Thulasiramanz P, Xuemin(Sherman) S (2010) Utility-Based Asynchronous Flow Control Algorithm for Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications* 28(7):1116–1126
5. Gartner. (2013) "Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020". <https://www.gartner.com/newsroom/id/2636073>, 12 December 2013. Retrieved 2 Jan 2014.
6. Liu Y, He Y, Li M, et al. (2013) Does wireless sensor network scale? A measurement study on GreenOrbs[J]. *IEEE Transactions on Parallel and Distributed Systems* 24(10):1983–1993
7. Tunca C, Isik S, Donmez MY, et al. (2014) Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey[J]. *IEEE Communications Surveys & Tutorials* 16(2):877–897
8. Zhang Y, He S, Chen J (2016) Data Gathering Optimization by Dynamic Sensing and Routing in Rechargeable Sensor Networks. *ACM/IEEE Transactions on Networking*. doi:10.1109/TNET.2015.2425146 to appear
9. Meng W, Wang X, Liu S (2016) Distributed load sharing of an inverter-based microgrid with reduced communication. *IEEE Transactions on Smart Grid*. doi:10.1109/TSG.2016.2587685
10. J. Ren, Y. Zhang, K. Zhang, and A. Liu. (2015) Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks. *IEEE Transactions on Industrial Informatics*, pages 1–1.
11. Wu X, Chen G, Das SK (2008) Avoiding energy holes in wireless sensor networks with nonuniform node distribution. *IEEE Transactions on Parallel and Distributed Systems* 19(5):710–720
12. He L, Pan J, Xu J (2013) A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements. *IEEE Trans Mob Comput* 12(7):1308–1320
13. Sasaki Y, Nakayama H, Ansari N, Nemoto Y, Kato N (2008) A new data gathering scheme based on set cover algorithm for mobile sinks in wsns. *IEEE GLOBECOM 2008*:1–5
14. Ma M, Yang Y, Zhao M (2013) Tour planning for mobile datagathering mechanisms in wireless sensor networks. *IEEE Trans Veh Technol* 62(4):1472–1483

15. J. Luo and J. P. Hubaux. (2005) Joint mobility and routing for lifetime elongation in wireless sensor networks. In Proceedings - IEEE INFOCOM, pages 1735–1746.
16. P. Gupta and P. R. Kumar. (1998) Critical power for asymptotic connectivity in wireless networks. Systems and Control Foundations and Applications, pages 547–566.
17. Xing G, Wang T, Jia W, Li M (2008) Rendezvous design algorithms for wireless sensor networks with a mobile base station. Proceedings of ACM Mobihoc 5317(6):231–239
18. Liu W, Fan J, Zhang S, Wang X (2013) Relay hop constrained rendezvous algorithm for mobile data gathering in wireless sensor networks. Lect Notes Comput Sci 8147:332–343
19. Liu W, Lu K, Wang J, Xing G, Huang L (2012) Performance analysis of wireless sensor networks with mobile sinks. IEEE Trans Veh Technol 61(6):2777–2788
20. Yuan Y, Peng YX, Shan-Shan LI, Tang WS (2011) Efficient heuristic algorithm for the mobile sink routing problem. Journal on Communications 32(10):107–117
21. Akkaya K, Younis M, Bangad M (2005) Sink repositioning for enhanced performance in wireless sensor networks. Comput Netw 49(4):512–534
22. Đurišić MP, Tafa Z, Dimić G, et al. (2012) A survey of military applications of wireless sensor networks[C]. Embedded. Computing:196–199
23. Leccese F, Cagnetti M, Calogero A, et al. (2014) A new acquisition and imaging system for environmental measurements: an experience on the Italian cultural heritage.[J. Sensors 14(14):9290–9312
24. Kennedy J, Eberhart R (1995) Particle swarm optimization. IEEE International Conference on Neural Networks, 1995 Proceedings 4: 1942–1948
25. Wang J, Chen X, Fang D, et al. (2015) Transferring compressive-sensing-based device-free localization across target diversity[J. IEEE Trans Ind Electron 62(4):2397–2409
26. Liu C, Fang D, Yang Z, et al. (2016) RSS distribution-based passive localization and its application in sensor networks[J. IEEE Trans Wirel Commun 15(4):2883–2895



**Wei Wang** received her B.S. and M.S. degrees in 2000 and 2004 from Xidian University, Xi'an, China, in Communications and Information System. She is studying for her PhD degree in Northwestern Polytechnical University, Xi'an, China. She is currently a lecturer at the School of Information Science and Technology, Northwest University. Her research interests include routing, topologic control and localization algorithm of wireless networks.