Energy-efficient routing for mobile data collectors in wireless sensor networks with obstacles

Energy-Efficient Routing for Mobile Data Collectors in Wireless Sensor Networks with Obstacles

Guangqian Xie · **Kaoru Ota** · **Mianxiong Dong** · **Feng Pan** · **Anfeng Liu**

Abstract This paper proposes an energy-efficient routing mechanism by introducing intentional mobility to wireless sensor networks (WSNs) with obstacles. In the sensing field, Mobile Data Collectors (MDCs) can freely move for collecting data from sensors. An MDC begins its periodical movement from the base station and finally returns and transports the data to the base station. In physical environments, the sensing field may contain various obstacles. A research challenge is how to find an obstacle-avoiding shortest tour for the MDC. Firstly, we obtain the same size grid cells by dividing the network region. Secondly, according to the line sweep technique, the spanning graph is easily constructed. The spanning graph composed of some grid cells usually includes the shortest search path for the MDC. Then, based on the spanning graph, we can construct a complete graph by Warshall-Floyd algorithm. Finally, we present a heuristic tour-planning algorithm on the basis of the complete graph. Through simulation, the validity of our method is verified. This paper contributes in providing an energy-efficient routing mechanism for the WSNs with obstacles.

1 Introduction

Advances in microelectronics technology, computer technology and wireless communication technology have led to the development of low-power and low-cost sensor nodes in the past few years. In the recent decade, Wireless Sensor Networks(WSNs) including these sensor nodes are becoming a hot research topic and have been applied in many respects including environmental monitoring, health monitoring, military surveillance, and many others as Internet of Thing(IoT) [1-5]. However, power supplies for sensor nodes are limited and hard to replace. In addition, nodes near the base station consume more energy than those elsewhere, since they relay the data collected by sensor nodes far away from the base station. Hence, once these sensors near the base station fail, the data collected by other sensors cannot be transferred to the base station. Then, the entire network becomes disconnected, although most of the nodes can still have a lot of energy. Therefore, to prolong the lifetime of the WSNs, minimizing the energy consumption of sensor nodes is the key challenges for WSNs.

Recent work shows that we can use mobile nodes to reduce the energy cost of WSNs to a large extent. Compared with static nodes, mobile nodes have more energy and more powerful capabilities. The data from static nodes can be collected by mobile nodes in one-hop or multi-hop way. At the same time, mobile nodes can move around the sensing

G. Xie (✉)
Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University,Wuxi, China
Changzhou Institute of Technology, Changzhou, China
E-mail: xgqmail2000@163.com

K. Ota
Department of Information and Electronic Engineering,Muroran Institute of Technology, Japan
E-mail: ota@csse.muroran-it.ac.jp

M. Dong (✉)
Department of Information and Electronic Engineering,Muroran Institute of Technology, Japan
E-mail: mx.dong@ieee.org

F. Pan
Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University,Wuxi, China
E-mail: xiegq@czu.cn

A. Liu
School of Information Science and Engineering, Central South University, Changsha, China
E-mail: afengliu@mail.csu.edu.cn

field. The papers [6-9] have proposed several different approaches. In this paper, mobile nodes are used as the Mobile Data Collectors (MDCs) which move through the sensing field. In addition, we assume that MDCs have unlimited energy. On the one hand, MDCs can be used as ferries, which carry data to the base station for static sensors so that expensive multi-hop communication and long distance communication are reduced. On the other hand, MDCs serve as a mobile "data transporter" through every region and link separate sub-networks together. Therefore, we can use MDCs to extend the network lifetime. In physical environments, the sensing field could contain various obstacles. For example, in fire detection systems, numerous sensor nodes are deployed on a forest area where various obstacles exist. MDCs, which are placed on a motorized vehicle or carried by a human, could move through the forest to find an obstacle-avoiding shortest tour for collecting data generated by sensor nodes if the terrain is not suitable for navigation. MDCs carried by fire-fighters would assist them for extinguishing the forest fire by providing them with enough information about the area of interest. To prolong the network lifetime, a research challenge is how to find an obstacle-avoiding shortest tour for the MDC.

In this paper, we assume that MDCs are low-rate for collecting data from static sensors. In addition, to upload data to the base station, MDCs are assumed to be delay-insensitive for accumulating data into fixed-size packets. Here, we assume that WSNs can tolerate some extent of delay. The MDC begins its periodical movement from the base station and finally returns for transferring the data to the base station. Once its moving path is planned, the MDC can move near the sensor nodes and consume less energy. Hence, we can greatly extend the network lifetime. Here, we assume that the network lifetime is the time interval from the static sensors start sending data to the MDC until the death of all static sensors. Current research has addressed the scheduling for MDCs in WSNs. However, a research challenge is how to efficiently dispatch the MDCs to find an obstacle-avoiding shortest tour in the presence of obstacles. In particular, once the sensing field contains obstacles of any shape and size, the deployment of MDCs will become more complex. To dispatch MDCs in WSNs with obstacles, we propose three steps as follows:

1) Grid Cells for Region Division. We divide the sensing region into the same size grid cells in which there may be a different number of static sensors. Grid cells are considered to be the basic unit and their size is closely related to communication radius of static sensors in this paper. MDCs can collect all sensing data from static sensors inside the same grid cell. In order to dispatch MDCs, we assume that there is one position (called sink position) inside each grid cell, where MDCs can move to the position for collecting data from static sensors. The sink position is usually located in the geometric center of the grid cell and it is represented as a square in this paper. Except the location of obstacles, MDCs can move to any position.

2) Obstacles Shape Regularization. In fact, the sensing field may contain various obstacles. How to efficiently dispatch MDCs to find an obstacle-avoiding shortest tour is always a research challenge. When the sensing region is divided into the same size grid cells, obstacles will contain some grid cells. Edges of obstacles intersect grid cells and obstacles may occupy part of some grid cells. Once obstacles occupy part of one grid cell, we assume that the grid cell is regarded as obstacles. Therefore, we obtain regularization shape of obstacles so that scheduling for MDCs becomes easier. MDCs usually have a large data communication radius. Hence, static sensors are located in the grid cells regarded as obstacles and their sensing data can be collected by MDCs. Taking into account the complexity of the system, this paper just considers to dispatch one MDC for the obstacle-avoiding shortest tour.

3) Spanning Graphs Application. Considering the existence of obstacles in WSNs, our goal is to find an obstacle-avoiding shortest tour by an efficient method. It is obvious that the shape of obstacles becomes a polygon after regularization. To find an obstacle-avoiding shortest tour, we construct a spanning graph by applying the sweep line algorithm proposed in [10]. Based on the spanning graph, we can acquire a complete graph by the Warshall-Floyd algorithm. With the search space of the MDC from all grid cells to the spanning graph obtaining grid cells, the scheduling for the MDC will become more efficient. By applying the heuristic tour-planning algorithm, we can find an obstacle-avoiding shortest tour on the basis of the complete graph.

Our main contributions are the following:

- We obtain the regularization shape of obstacles by the grid-based technique that makes the scheduling for MDCs easier.
- We construct a spanning graph by applying the sweep line algorithm for Wireless Sensor Networks with Obstacles.
- We provide the heuristic tour-planning algorithm by which we can find an obstacle-avoiding shortest tour.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 introduces the network model and the problem formulation. We propose a heuristic tour-planning algorithm to finding an obstacle-avoiding shortest tour in Section 4. Section 5 proposes the evaluation of our technique. Finally, we conclude the paper with future works in Section 6.

2 Related Works

Recent work shows that we can utilize the mobility of nodes to reduce the energy cost of WSNs to a large extent. Many papers have proposed several different approaches. Mobility is thought to ease the traffic burden and enhance energy efficiency in WSNs. We then study the relevant techniques of the mobility of nodes in the literature.

In [11], the authors propose a new data-gathering mechanism for WSNs by introducing the mobile data collector (M-collector) into the network. An M-collector, which works like a mobile base station, is introduced to collect sensing data from static sensors. The MDC begins its periodical movement from the base station and finally returns for transferring the data to the base station. For some applications in large-scale networks, the authors take a divide-and-conquer strategy and use multiple M-collectors, each of which moves through a shorter data-gathering tour. According to the wireless energy transfer technology, the authors present a formal optimization framework by jointly optimizing traveling path, flow routing, and charging time in [12]. The Wireless Charging Vehicle (WCV) starts a periodical tour from the service station, traverses the network for charging some sensor nodes wirelessly, and finally returns. According to the novel Reformulation-Linearization Technique (RLT), the authors design a near-optimal solution for the optimization problem. However, in [11, 12], the authors dont consider that the sensing field may contain various obstacles. In fact, the tour for mobile nodes in sensing field containing obstacles is more complex than that sensing field without obstacles. In [13, 14], the authors present a two-phase dispatch heuristic for the multi-round multi-attribute sensor dispatch problem. In a hybrid WSN, static sensors monitor and collect environment information. Once events happen, each static sensor can only sense one attribute of events. Compared with static sensors, a mobile sensor can evaluate multiple attributes of events. According to the sensing data from static sensors, mobile sensors move to corresponding hot locations for more in-depth analysis. To minimize the energy consumption, the authors present a two-phase heuristic algorithm to dispatch mobile sensor for hot locations. In the first phase, the authors dispatch MAM sensors to hot locations in a one-to-one approach. In the second phase, according to unassigned hot locations, the authors present a spanning-tree construction algorithm for the displacement of MAM sensors. Due to similar capabilities of sensors, a research challenge is how to dispatch mobile sensors to these hot locations. The paper [15] proposes a load balancing algorithm which tries to balance the number of sensor nodes each mobile element services. If the network scale is large, multiple of data mules will be required. Sensor nodes are usually freely configurable in practical circumstances. In particular, the use of load balancing is considered as far as possible. The author presents a load balancing algorithm for the data mules. However, data mules cannot always move along straight lines, for instance, the moving paths of data mules can be blocked by obstacles or boundaries. In [16], the authors present an algorithm for finding a route for the mobile base station that visits all the nodes in the network. The main idea of the algorithm is to divide nodes into bins according to their geographic location and their overflow time. In each bin, a route that visits each node once is computed using a solution for the Traveling Salesperson Problem (TSP). These routes are then concatenated to form a schedule for visiting all the nodes in the network. Once this schedule is found, the speed of the base station is adjusted to make the route feasible. In [17], the authors present a detailed review of the routing protocols that serve for the distributed mobile sink. At the same time, the authors discuss the unique challenges between mobile sinks and the design requirement of their routing protocol. Then, the motivations and the inherent mechanisms are provided. Finally, the authors propose a classification of the protocols whose advantages and drawbacks are individually determined from the performance requirements.

In [18, 19], the authors present a simple way to dispatch the mobile sensor to the event location in Wireless Sensor Networks with Obstacles. Once mobile sensors receive the sensing data from the static sensors, they will move to these hot locations to conduct further analysis. A research challenge is how to find an obstacle-avoiding shortest path for the mobile sensor to the hot location. The authors propose modified Dijkstras algorithm to solve scheduling for the mobile sensor in the presence of obstacles. In fact, in [18, 19], the authors just consider that the shape of obstacles is convex. However, scheduling for mobile sensors in sensing field containing concave obstacles is more complex than that in sensing in sensing field containing convex obstacles.

In summary, current research has addressed some scheduling problem of mobile nodes in WSNs from different aspects. However, as discussed above, to prolong the network lifetime, a research challenge is how to find a shortest tour for mobile nodes. In particular, once the sensing field contains obstacles of any shape and size, the deployment of mobile sensors will become more complex. This paper aims to presents a solution to this issue.

3 Network Model and Problem Formation

In this section, we propose a network model in this research and problem formulation for this paper.

3.1 Network Model

Wireless Sensor Networks use sensor nodes to detect information from physical environments. In the past, the information from the environment can only be collected by static nodes in WSNs. With the recent development of technology, we can utilize mobile nodes to collect information. Therefore, the network performance is further improved in many ways. Static sensors can be randomly configured and the region coverage of the WSNs is one of the most important problems. In this paper, we assume that no hole exists in the WSN and static sensors are the same in their capabilities. In physical environments, the sensing field may contain various obstacles. A research challenge is how to find an obstacle-avoiding shortest tour for the MDC in the presence of obstacles. The Mobile Data Collector (MDC) is a mobile device, which could be a vehicle or a mobile robot with enough energy. Here, the network lifetime is the time interval from the static sensors start sending data to the MDC until the death of all static sensors.

In this paper, we present an efficient scheduling technique for the MDC in the presence of obstacles, by which we can prolong the lifetime of the WSNs as long as possible. Taking into account the complexity of the system, we assume there is only one MDC in the WSN. We consider N sensor nodes distributed over a plane with M obstacles. Referring to Figure 1, static sensors are deployed unevenly in the network, denoted by pentagrams. At the same time, there are four black irregular figures, which represent obstacles. We assume that the base station is located in top-left corner of the two-dimensional area. The MDC begins its periodical obstacle-avoiding movement from the base station and finally returns for transferring the data to the base station. Once its moving path is planned, the MDC can move near the sensor nodes and consume less energy. Hence, we can greatly extend the network lifetime.

3.2 Problem Formulation

In the following paragraphs, we present a problem formulation described above. We consider wireless sensor networks in which sensor nodes are freely deployed in a plane with various obstacles. The sensing data from the surrounding environment is collected by sensor nodes. Once the MDC moves near sensor nodes, these sensor nodes will send the collected data to it. Except the location of obstacles, we assume that the MDC can traverse the sensing field and collect the sensing data from the sensor nodes at certain locations. We assume that the set of locations Γ is denoted by $\Gamma = \{\ell_0, \ell_1, \ell_2, \cdots, \ell_{n_l}\}$, where ℓ_0 represents the starting position and the ending position of the obstacle-avoiding tour. Let S be the set of sensor nodes. When the MDC is at $\ell \in \Gamma$,
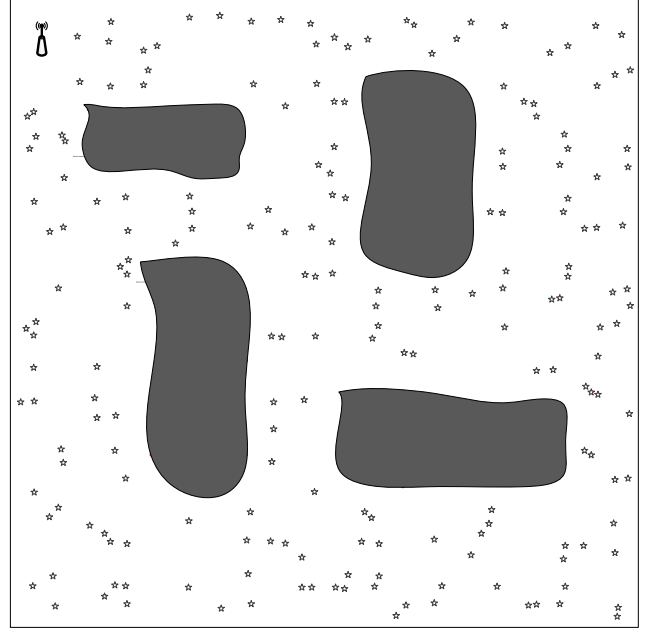

Fig. 1 An example of scheduling for the MDC with obstacles

the neighbors of node i is denoted as

$$NB(i,\ell) = \{ j \in S \cup \{\ell\} \mid D(i,j) \le \overline{D}, j \neq i \} \tag{1}$$

Here, D(i,j) is the distance between node i and j, and $\overline{D}$ is the communication range of the sensor nodes. The model of the obstacle-avoiding tour is formulated as follows [11, 20].

$$Minimize \sum_{i,j \in \Gamma, i \neq j} D(i,j)\delta_{ij} \tag{2}$$

$$s.t. \sum_{i \in \Gamma, i \neq j} \delta_{ij} = I_j, \forall j \in \Gamma \tag{3}$$

$$\sum_{j \in \Gamma, j \neq i} \delta_{ij} = I_i, \forall i \in \Gamma \tag{4}$$

$$\sum_{j \in NB(i,\ell)} I_i \ge 1, \forall j \in S \tag{5}$$

$$\sum_{j \in NB(i,\ell)} x_{ij}^{(\ell)} - \sum_{k:i \in NB(k,\ell)} x_{ki}^{(\ell)} = d_i, \forall i \in S, \forall \ell \in \Gamma \tag{6}$$

$$x_{ij}^{(\ell)} \ge 0, \forall i \in S, \forall \ell \in \Gamma, j \in NB(i,\ell) \tag{7}$$

where

$$\delta_{ij} = \begin{cases} 1, \text{if the obstacle-avoiding tour contains path } D(i,j) \\ 0, \text{otherwise} \end{cases}$$

$$I_i = \begin{cases} 1, \text{if the obstacle-avoiding tour contains locations } \ell \in \Gamma \\ 0, \text{otherwise} \end{cases}$$

d_i : node i obtains data from physical environments;
$x_{ij}^{(\ell)}$:the rate assignment from node i to the node j when the MDC is at $\ell \in \Gamma$.

In order to find an obstacle-avoiding shortest tour, the optimization problem (2) minimizes the total moving path for the MDC. Constrain (3) ensures that there must be one arc pointing to one node in the obstacle-avoiding shortest tour. Compared with constrain (3), except the arc pointing to one node, constrain (4) ensures that the other arc is away from it. Constraint (5) assures every sensor in the neighbor set that no less than one location belongs to the moving path. Thus, each sensor can directly communicate with the MDC. Condition (6) ensures that all nodes are the flow conservation when the MDC stops at $\ell \in \Gamma$. It is obvious that the obstacle-avoiding shortest tour is NP-hard. To solve the obstacle-avoiding shortest tour problem, we will propose a heuristic tour-planning algorithm in the next section.

4 Heuristic Tour-planning Algorithm

In this section, we propose a heuristic tour-planning algorithm for the MDC to find the obstacle-avoiding shortest tour. Firstly, we will present a grid-based technique to facilitate the dispatch of the MDC. Secondly, by applying the line sweep technique, we can construct the spanning graph of the network model. Then, based on the spanning graph, we can acquire a complete graph by Warshall-Floyd algorithm. Finally, the obstacle-avoiding shortest tour is acquired by using the heuristic algorithm. Specific steps for heuristic algorithm are described as follows:
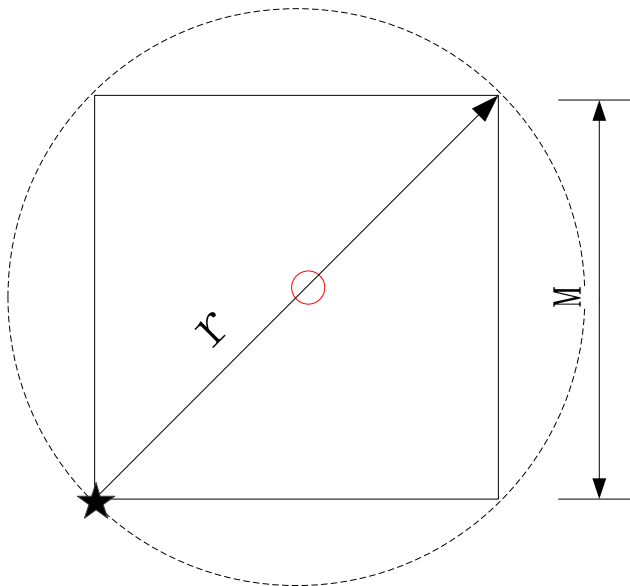

Fig. 2 The sense range *r* of the static sensor node and the grid size *M*

4.1 Dividing Network Region into Grid Cells

In order to simplify the dispatch of the MDC in Figure 1, we first divide the whole region into many grids, each of equal size. In this paper, the division of the grid is based on the sense range of the sensor node. Here, we assume that there is long enough communication range for the static sensor node. The papers [21-24] use grid-based techniques to analyze and solve problems in WSNs. With this method, the sensing field is divided into the fixed number of equal size grids. Without loss of generality, we assume that there is no hole in the WSN and static sensors are the same in their capabilities. Therefore, each grid contains at least one sensor in it. In order to calculate the grid size, we take into account the special circumstances that there is only one sensor inside the grid. In Figure 2, the static sensor node is denoted by the pentagram and located in the lower left corner. Let *M* be the grid size. At the same time, we assume that *r* is the sense range of nodes. According to Figure 2, if no hole exists in the WSN, the size of the grid is calculated as follows.

$$r^2 = M^2 + M^2 \implies M = \frac{r}{\sqrt{2}} \qquad (8)$$

From the equation (8), we can declare the relations between the sense range *r* of nodes and the grid size *M*. There will be no hole in the WSN when the grid size *M* equals $\frac{1}{\sqrt{2}}$ times the sense range *r* of nodes.
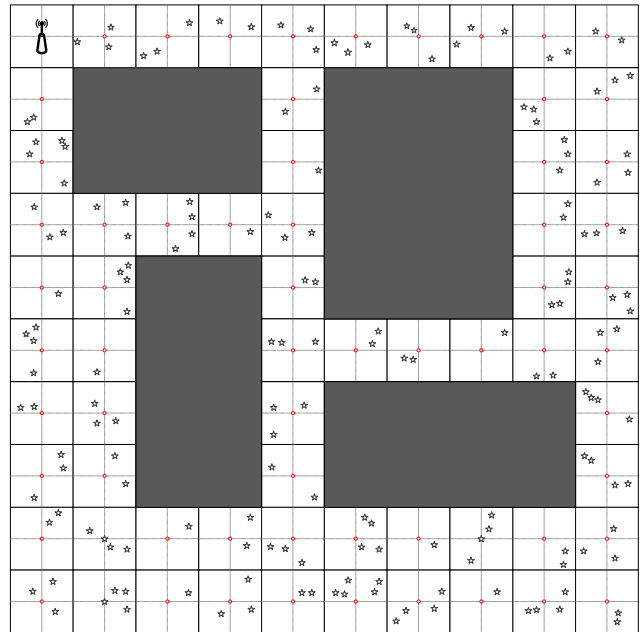

Fig. 3 The grid graph of monitored region

In this paper, there is one position called sink position in each grid cell where the sink position is located in the geometric center. Once the MDC stays at the sink position in one grid cell, it can collect all sensing data from sensors

inside the grid cell. In Figure 2, the sink position is denoted by the red circle. Therefore, when the MDC stays at the red circle position, it can collect sensing data from all static sensors (denoted by the pentagram) inside the grid cell.

By using the technique of the grid division in Figure 2, Figure 3 shows the grid division of the monitored region in Figure 1. Obviously, four obstacles contain some grid cells and their edges intersect grid cells. Once obstacles occupy part of one grid cell, the grid cell is regarded as obstacles in this paper. Therefore, the regularization shape of obstacles is acquired by grid-based techniques. Based on this method, we obtain four regularization shapes of the obstacles in Figure 3. Red circles and pentagrams represent sink positions of each grid cell and static sensors, respectively. In this paper, the base station is located in top-left corner in the two-dimensional area. Next we present the method based on spanning graphs for the MDC.

4.2 Spanning Graph Construction

On a basis of the minimum spanning tree, we can find an obstacle-avoiding shortest tour for the MDC. It is proved in [17] that the minimum spanning tree can be obtained from the spanning graph. In this section, we will discuss how to construct the spanning graph. Several studies have addressed the spanning graph construction [10, 25]. A spanning graph is usually an undirected graph, by which the infinite possible locations for the MDC movement can be reduced to a finite set of locations. The obstacle-avoiding spanning graph can be formed by making connections between terminals and obstacle corners. Based on the sweep line algorithm [10], we can construct the spanning graph.
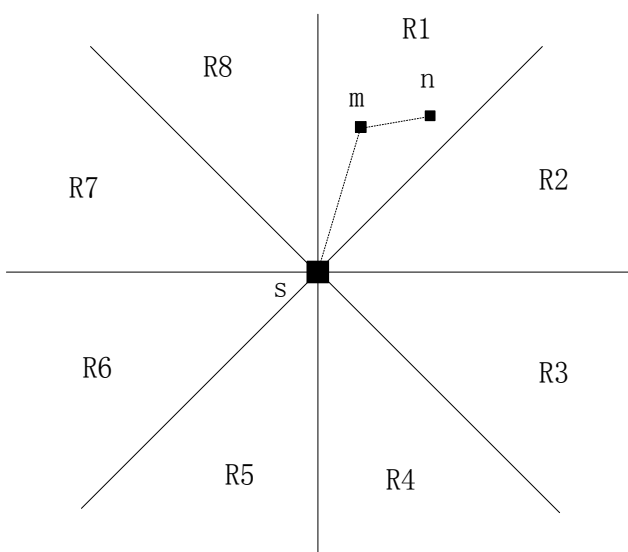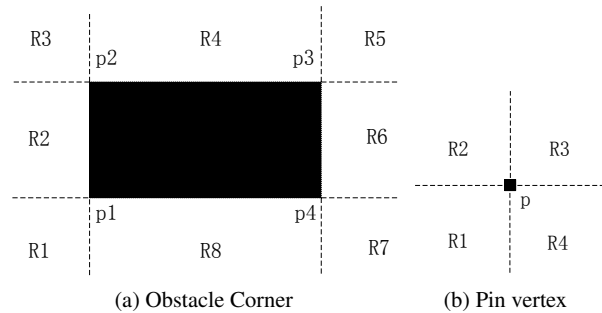

Fig. 4 Eight regions of a terminal

Fig. 5 Search regions for blockage and pin

Figure 4 shows eight regions of a terminal, by which we illustrate the construction of the spanning graph. Referring to Figure 4 the point s is defined as coordinate origin, through which the plane is divided into 8 regions ($R1 \sim R8$) by different direction lines. Compared with the maximum distance from the point s to one of any two points in one region, the distance between these two points is always small, which has been proved. A minimal spanning tree is acyclic, so the maximum distance from s to one of any two points in one region is not included in the minimal spanning tree. According to the above analyses, it is obvious that the minimal spanning tree must obtain the segment line consisted of the point s and the point near the point s. For example, in Figure 4, the longest edge (s,n) is not contained in the minimal spanning tree. In order to simplify the calculations, we divide a plane into 4 regions in this paper. It is proved in [25] that the quadrant partition, compared with the octant partition, also exhibits good results.

Figure 5 describes the quadrant partition for an obstacle corner and a pin vertex, respectively. By this method for processing the grid graph in Figure 3, we obtain the spanning graph in Figure 6. The algorithm of the spanning graph construction is summarized in Algorithm 1. The spanning graph construction is performed by using edge connection for different region of all pin vertices and all obstacle corners. Once Algorithm 2 is called in the region $R2$ and $R6$ of all obstacle corners, the corresponding edges will join the spanning graph (lines 1-2). Here, we sort elements of the list V by non-decreasing x coordinates. The line sweep sequence is reverse between the region $R2$ and $R6$, so the same sorted list V can be applied by changing the line sweep direction. Likewise, we also call Algorithm 2 in the region $R4$ and $R8$ of all obstacle corners. However, unlike sorted list V in the Region $R2$ and $R6$, we sort elements of the list V by non-decreasing y coordinates in the region $R4$ and $R8$ of all obstacle corners (lines 3-4). Besides the region $R1$ and $R5$ of all obstacle corners, the region $R1$ and $R3$ of all pin vertices is also added to our search region (lines 5-6). We sort elements of the list V by non-decreasing $x+y$ coordinates in the region $R1$ and $R5$ of all obstacle corners and the region $R1$ and $R3$ of all pin vertices. To construct the spanning graph,

we call Algorithm 2. Likewise, we also call Algorithm 2 in the region $R3$ and $R7$ of all obstacle corners and the region $R2$ and $R4$ of all pin vertices. Here, the elements of the list V are sorted by non-decreasing $y - x$ coordinates (lines 7-8). According to the quadrant partition for an obstacle corner and a pin vertex, we finally construct the spanning graph. Consequently, with the search space of the MDC from the grid graph to the spanning graph, the possible locations for the MDC movement can be greatly reduced. Therefore, our algorithm makes it more efficient for the MDC to find an obstacle-avoiding shortest tour.
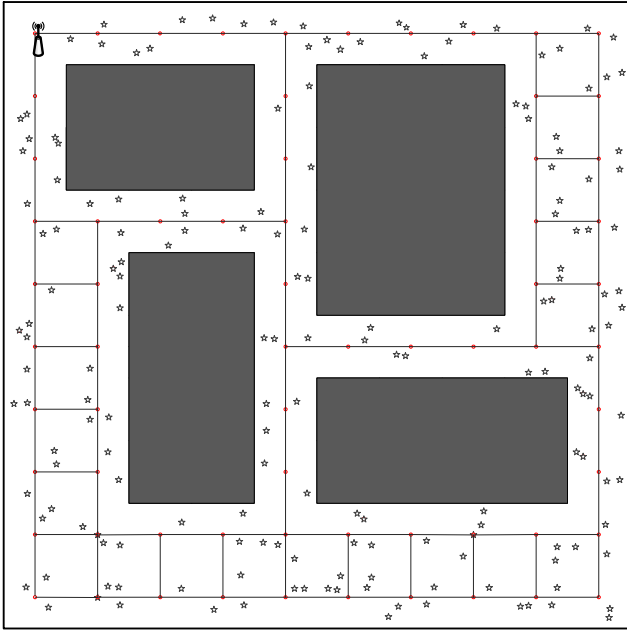

Fig. 6 The spanning graph based on grid division

Algorithm 1: Spanning graph construction

    input : the set of V nodes of pin vertices and obstacle corners
    output: the spanning graph G for V

1 sort V by non-decreasing x;
2 call **Algorithm 2** in region $R2$ and $R6$ of all obstacle corners;
3 sort V by non-decreasing y;
4 call **Algorithm 2** in region $R4$ and $R8$ of all obstacle corners;
5 sort V by non-decreasing $x + y$;
6 call **Algorithm 2** in region $R1$ and $R5$ of all obstacle corners and in region $R1$ and $R3$ of all pin vertices ;
7 sort V by non-decreasing $y - x$;
8 call **Algorithm 2** in region $R3$ and $R7$ of all obstacle corners and in region $R2$ and $R4$ of all pin vertices ;

Algorithm 2 presents the construction of the edge connection graph in different search regions. According to the corresponding coordinates, we sort all points in V_s by non-decreasing order. In order to applying the sweep line algo-

rithm, we need an active set for one point that knows all points in the set. This is a premise for using the sweep line algorithm. Consequently, we create an active set N where all points can be dynamically added and deleted. We initialize the active set N to an empty set and decide if each point s_i is in the sorted list V_s (lines 1-2). Then, we will add s_i into set N if s_i is not a corresponding location of a blockage (lines 3-4). Conversely, if s_i is a corresponding location of a blockage, the edge connection for s_i will be performed (lines 5-13). Here, we assume s_i is one of the four corners of the blockage b_j. Firstly, we choose the point s_{i+1} from list V_s, where the point s_i locates front-left of the point s_{i+1}. Secondly, we decide if each point s_i is in the an active set N. Once the point s_i locates in the corresponding region of b_j, the point s_i is added to the set N_k. If the blockage b_j is not a vertical blockage, we will delete points from N which are located in quadrant partition of s_i. Then, the point m closest to s_i and the point n closest to s_{i+1} are selected. Finally, we get two edges (s_i, m) and (s_{i+1}, n) which will be added to the spanning graph G. Note the point m could coincide with the point n in the set N_k. We deal with the next point in list V_s after the set N_k is vacated.

Algorithm 2: Edge connection graph construction

    input : the sorted list V_s with non-decreasing corresponding coordinates
    output: the edge connection graph G_s for quadrant partition

1 $N \leftarrow \varnothing$;
2 **for** *each $s_i \in V_s$* **do**
3   **if** *(s_i is not a corresponding location of blockage b_j)* **then**
4   add s_i to N;
5   **else**
6   **if** *($N! = \varnothing$)* **then**
7           $N_k = \varnothing$;
8   **if** *(b_j is not a vertical blockage)* **then**
9   delete points in quadrant partition of s_i which are from N;
10   add these points located in quadrant partition of s_i to N_k;
11   **if** *$N_k! = \varnothing$* **then**
12   choose the points m and n from N_k which are closest to s_i and s_{i+1}, respectively;
13   add the edges (s_i, m) and (s_{i+1}, n) to the edge connection graph G_s;

4.3 Complete Graph Construction

In this section, on the basis of the spanning graph, we construct the complete graph for the MDC to find an obstacle-avoiding shortest tour. In essence, the obstacle-avoiding shortest tour problem is similar to the Traveling Salesman Prob-

lem (TSP) which is a classical problem. The TSP aims to find a minimal weight tour where all the nodes of a weighted complete graph are included. In particular, each nodes of the weighted complete graph is traversed once and only once with the least cost. Note that nodes of the TSP can be regarded as the position where the MDC collects data from the static sensors. It has been proven that the TSP is a typical NP-hard problem and NP-complete problem. Because of its theoretical and practical values, the TSP has become a hot field in the theory of combinatorial optimization, where the most active research area is the exact or approximate algorithm design of the TSP. Due to the complexity of the TSP, we adopt an approximate algorithm. One of the main assumptions regarding the TSP is that the input is a complete graph. However, the spanning graph that we have constructed is not a complete graph. Therefore, figuring out the process on how to construct the complete graph is the key for the MDC to find the obstacle-avoiding shortest tour.

To better describe the algorithm, we must know some frequently used terms, which are defined as shown below. Note that common concepts in graph theory are not defined here. The paper [26] provides these common concepts.

Definition 1: A complete graph K_N is a graph in which there are N vertices and an edge between every two vertices.

Definition 2: A Hamilton circuit is a circuit in which every vertex of a graph is used only once.

Definition 3: A weighted graph is a graph in which a weight is assigned to each edge (denoted by the distance, time, etc).

Definition 4: The Traveling Salesman Problem (TSP) is the problem in which a minimum-weight Hamilton circuit in K_N is found.

According to these definitions, we can understand the essence of the problem better. Computing a complete graph can be done with the following method. For each vertex $n \in N$, we can compute a shortest path tree rooted at n by the shortest path algorithm. On the basis of the spanning graph obtained above, we can construct a complete graph by the Warshall-Floyd algorithm in this paper. Note that the distance between every two vertices in Definition 1 is the shortest path of these two vertices in the spanning graph, so they may not be directly connected.

Algorithm 3: Heuristic tour-planning algorithm

    input : the spanning graph G
    output: an obstacle-avoiding shortest path L_{path}

1   construct the complete graph $G' = (V, E')$ from the spanning graph $G = (V, E)$ by using the Warshall-Floyd algorithm ;
2   find a sufficient number of Hamilton circuits from the complete graph $G' = (V, E')$;
3   optimize these Hamilton circuits by the principle of one by one revision of two sides;
4   Choose the Hamilton circuit G_H with the smallest total weight;

The formula $G = (V, E)$ describes a spanning graph with n vertices, in which $V = \{v_1, v_2, \cdots, v_n\}$ are the vertices sets and $E = [e_{ij}]_{n \times n}$ are the edges matrix. In particular, the point $v_i(1 \leq i \leq n)$ is the vertex and $e_{ij}(1 \leq i, j \leq n)$ is the edge linking the vertices v_i and v_j. The edges matrix $E = [e_{ij}]_{n \times n}$ are assigned to corresponding weighs matrix $W = [w_{ij}]_{n \times n}$ and e_{ij} is equal to w_{ij}. Algorithm 3 presents a heuristic tour-planning algorithm for the MDC to find an obstacle-avoiding shortest tour. Firstly, Let the weight $w(x, y)$ for $\forall (x, y) \in E'$ in the complete graph $G' = (V, E')$ is equal to the shortest path between vertex x and vertex y in the spanning graph $G = (V, E)$ (line 1). Then, a large number of Hamilton circuits with the weighs are optimized (lines 2-3). Finally, we obtain the obstacle-avoiding shortest path L_{path}, i.e., the Hamilton circuit with the smallest total weight(line 4).

5 Evaluation

This section introduces the implementation of our approach. In addition, the proposed algorithms have been implemented using Matlab. We present the results from experiments which evaluate the technique proposed in previous. Simulation settings and Evaluation Results are as follows.
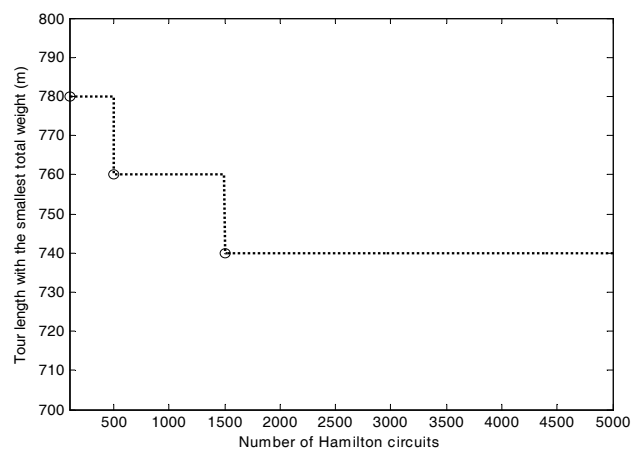

Fig. 7 Tour length comparison for a different number of Hamilton circuits

5.1 Simulation Settings

In this experiment, sensor nodes are deployed over a $100 \times 100 m^2$ field, which contains four obstacles. There are 170 sensor nodes which are randomly distributed over the field. The base station is at (5, 5) (in meters), where is also assumed to be at the origin of the MDC. The MDC begins its periodical tour from the position (5 m, 5 m) and finally returns to the same position. Figure 1 shows an example of

scheduling for the MDC with four obstacles. We assume that the sense range r of the static sensor node is fixed and equal to $10\sqrt{2}m$. According to the former algorithm, it is obvious that the grid size M equals $10m$. Therefore, we can partition the two-dimensional plane with $10 \times 10 = 100$ grid cells (see Figure 3). The radio characteristics used in our simulations are the same as these in [27].

In order to simplify scheduling for the MDC in the presence of obstacles, we assume that the data collected by static sensors is the delay-tolerant data, i.e., they will wait for the MDC to come and pick them up. In addition, the traveling speed of the MDC is fixed.

5.2 Evaluation Results

We have conducted numerous experiments to verify our algorithms in previous. With the increase of the number of Hamilton circuits from the complete graph, Figure 7 shows that tour length becomes shorter. However, the tour length with the smallest total weight will not change when the number of Hamilton circuits becomes larger, i.e., we obtain the obstacle-avoiding shortest path. At the same time, Figure 7 plots the performance of the tour length for values of the number of Hamilton circuits that vary between 100 and 5000. By the principle of one by one revision of two sides, we measured that the tour length with the smallest total weight is equal to 780m, 760m and 740m, when the number of Hamilton circuits is equal to 100, 500, and 1500, respectively. The results demonstrate that, by method based on heuristic algorithm for TSP, we can find an obstacle-avoiding shortest tour from a sufficient number of Hamilton circuits. Therefore, we acquire the energy-efficient routing for the MDC which begins its periodical tour from the position (5 m, 5 m) and finally returns to the same position. Figure 8 shows the movement path of the MDC in the spanning graph, when the number of Hamilton circuits is equal to 2000. The MDC moves to the sink position in each grid cell where it collects data sensed by static sensors. Here, we assume that the MDC collects data only once each round at the sink position.

Below, we compare our heuristic tour-planning algorithm with the LEACH [27]. In this paper, the lifetime of the network is relative and the MDC moves to the starting position at the beginning of each round. At the same time, we use $x\%$ network lifetime to represent the network lifetime. At the moment, $(100 - x)\%$ sensors have exhausted their energy. The operation of the LEACH scheme and our MDC scheme is based on rounds. Each round begins with data transferred and ends with all data transferred to the base station. Figure 9 shows the lifetime comparison between the LEACH scheme and our MDC scheme. According to the number of rounds, we evaluate $x\%$ network lifetime of two schemes. Referring to Figure 9, we observe that, by moving the MDC through a well-planned tour, the lifetime of our
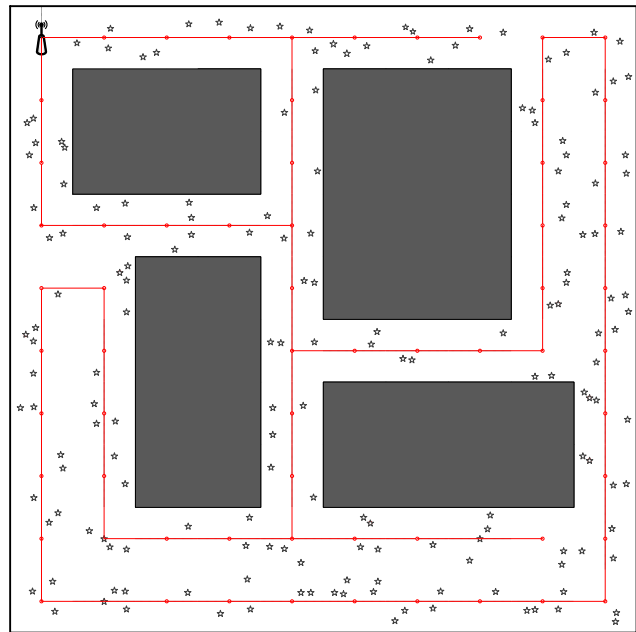

Fig. 8 The movement path of the MDC

MDC scheme can be prolonged significantly compared with the LEACH scheme. For example, the first sensor node in our MDC scheme dies when the number of rounds is equal to 2260, but the first one in the LEACH scheme dies when the number of rounds is equal to 922.
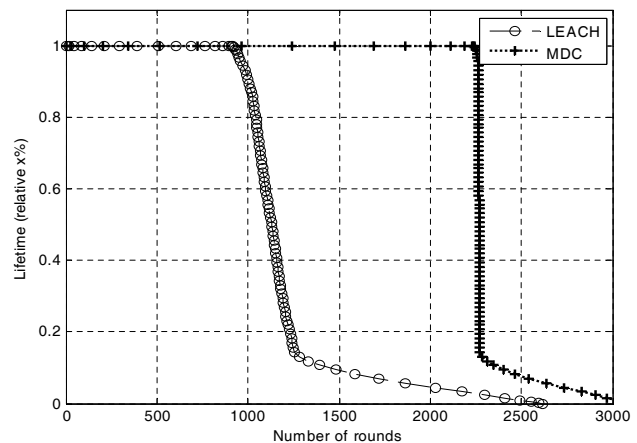

Fig. 9 Relative $x\%$ network lifetime

We have analyzed these experiments in tour length and lifetime. According to these experiments, the scheduling for the MDC scheme is more efficient than that for the LEACH scheme. By method based on our heuristic tour-planning algorithm, we can quickly find an obstacle-avoiding shortest tour. In addition, this also indicates that our technique is feasible for scheduling for the MDC.

6 Conclusions and Future Works

We have formulated the dispatch problem of the MDC in a WSN with obstacles in this paper. To efficiently schedule the MDC to find an obstacle-avoiding shortest tour, a method based on the heuristic tour-planning algorithm is developed. Firstly, we divide the network region into the same size grid cells and the MDC can move to the sink position inside each grid cell to collect sensing data. Secondly, to facilitate scheduling for MDC, regularization shape of obstacles is acquired by grid-based techniques. Finally, we find an obstacle-avoiding shortest tour based on spanning graphs. The experimental results show that our approach is feasible for the dispatch of the MDC and the network lifetime is prolonged. Once the search space is restricted to the spanning graphs, the MDC can be efficiently dispatched to find an obstacle-avoiding shortest tour. In this paper, we just consider the obstacle-avoiding shortest tour problem for one MDC. As for the future work, we plan on extending current work to the obstacle-avoiding shortest tour problem for many MDCs.

References

1. H. L. Fu, H. C. Chen and P. Lin. Aps: Distributed air pollution sensing system on wireless sensor and robot networks. Computer Communications, 35(9), 1141-1150(2012)
2. Z. Shen, X. Pan, C. Huang, J. Feng, Y. Zhao, M. Gao and L. M. Ni. Energy consumption monitoring for sensor nodes in snap. International Journal of Sensor Networks, 13(2), 112-120(2013)
3. J. Cuevas-Martinez, J. Canada-Bago, J. Fernandez-Prieto and M. Gadeo-Martos. Knowledge-based duty cycle estimation in wireless sensor networks: Application for sound pressure monitoring. Applied Soft Computing, 13(2), 967-980(2013)
4. M. Dong, X. Liu, Z. Qian, et al. QoE-Ensured Price Competition Model for Emerging Mobile Networks. IEEE Wireless Communications, 22(4), 50-57(2015)
5. B. Zhou, S. Yang, Thu Hien Nguyen, T. Sun and Kenneth T. V. Grattan. Wireless Sensor Network Platform for Intrinsic Optical Fiber pH Sensors. IEEE Sensors Journal, 14(4), 1313-1320(2014)
6. M. Dong, Kaoru Ota, Laurence T. Yang, et al. Mobile agent-based Energy-aware and User-centric Data Collection in Wireless Sensor Networks. Computer Networks (Elsevier), 74(B), 58-70(2014)
7. S.T. Guo, C. Wang and Y.Y. Yang. Joint Mobile Data Gathering and Energy Provisioning in Wireless Rechargeable Sensor Networks. Mobile Computing, IEEE Transactions on, 13(12), 2836-2852(2014)
8. M. Zhao, Y.Y. Yang and C. Wang. Mobile Data Gathering with Load Balanced Clustering and Dual Data Uploading in Wireless Sensor Networks. IEEE Transactions on Mobile Computing, 14(4), 770-785(2015)
9. L. J, Y. Y and W. C. Mobility assisted data gathering with solar irradiance awareness in heterogeneous energy replenishable wireless sensor networks. Computer Communications, 69, 88-97(2015)
10. H. Zhou, N. Shenoy and W. Nicholls. Efficient spanning tree construction without delaney triangulation. Information Processing Letter, 81(5), 271-276(2002)
11. M. Ma, Y.Y. Yang and M. Zhao. Tour Planning for Mobile Data-Gathering Mechanisms in Wireless Sensor Networks, IEEE Transactions on Vehicular Technology, 62(4), 1472-1482(2013)
12. L.G. Xie, Y. Shi, Y. Thomas Hou, et al. Multi-Node Wireless Energy Charging in Sensor Networks. IEEE/ACM Transactions on Networking, 23(2), 437-450(2015)
13. Y. C. Wang. A two-phase dispatch heuristic to schedule the movement of multi-attribute mobile sensors in a hybrid wireless sensor network. IEEE Transactions on Mobile Computing, 13(4), 709-722(2014)
14. Y. C. Wang. Efficient dispatch of multi-capability mobile sensors in hybrid wireless sensor networks. In IEEE VTS Asia PacificWireless Communications Symposium (APWCS)(2012)
15. D. Jea, A. A. Somasundara and M. B. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. in Proc. IEEE/ACM Int. Conf. DCOSS(2005)
16. Y. Gu, D. Bozdag and E. Ekici. Mobile element based differentiated message delivery in wireless sensor networks. In WoW-MoM(2006)
17. Can Tunca, Sinan Isik, M. Yunus Donmez and Cem Ersoy. Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey. IEEE Communications Surveys&Tutorials, 16(2), 877-897(2014)
18. Shwetha G. K, Sagarika Behera and Jithendranath Mungara. Energy-Balanced Dispatch of Mobile Sensors in Hybrid Wireless Sensor Network with Obstacles. IOSR Journal of Computer Engineering, 2(1), 47-51(2012)
19. Shalini Kumari H A and Shivanna K. Dispatch of mobile sensors in the presence of Obstacles Using Modified Dijkstra Algorithm. International Journal Of Computational Engineering Research, 2(5), 1458-1461(2012)
20. Y. Yun and Y. Xia. Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications. IEEE Transactions on Mobile Computing, 9(9), 1308-1318(2010)
21. C. Ai, L. Guo, Z. Cai and Y. Li. Processing area queries in wireless sensor networks. International Conference on Mobile Ad-hoc and Sensor Networks(2009)
22. I. H. Peng and Y. W. Chen. Energy consumption bounds analysis and its applictions for grid based wireless sensor networks. Journal of Network and Computer Applications, 36(1), 444-451(2013)
23. K. Xu, G. Takahara and H. Hassanein. On the robustness of grid-based deployment in wireless sensor networks. In Proceedings of the 2006 international conference on Wireless communications and mobile computing(2006)
24. Z. Zhou, J. Tang, L. Zhang, K. Ning and Q. Wang. EGF-tree: an energy-efficient index tree for facilitating multi-region query aggregation in the internet of things. Personal and Ubiquitous Computing, 18(4), 951-966(2014)
25. Zion Shen, Chris C.N. Chu and Ying-Meng Li. Efficient Rectilinear Steiner Tree Construction with Rectilinear Blockages. in Proceedings of the IEEE International Conference on Computer Design Conference(2005)
26. J. A. Bondy and U. Murthy. Graph Theory with Applications. New York: Elsevier(1976)
27. W.R.Heinzelman, A.P.Chandrakasan and H.Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, 1(4), 660-670(2002)
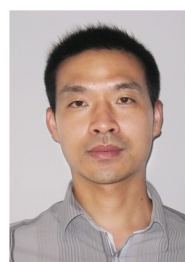
Guangqian Xie received M.S. degree in Control Theory and Control Engineering from Jiangnan University, China in 2004. He is now working toward the Ph.D. degree in Control Science and Control

Engineering, Jiangnan University, China. He is currently an Assistant Professor with College of Computer and Information Engineering, Changzhou Institute of Technology, China. From May 2015 to January 2016He was a visiting scholar at University of Ontario Institute of Technology, Canada. His research interests include wireless sensor networks, Internet of Things and intelligence control.

Kaoru Ota received M.S. degree in Computer Science from Oklahoma State University, USA in 2008 and Ph.D. degree in Computer Science and Engineering from The University of Aizu, Japan in 2012. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar with BBCR group at University of Waterloo, Canada. Also she was a Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. She has joined JSPS A3 foresight program as one of primary researchers since 2011 which is supported by Japanese, Chinese and Korean government. Dr. Otas research results have been published in 90 research papers in international journals, conferences and books. She is the Best Paper Award Winner of ICA3PP 2014, GPC 2015 and IEEE DASC 2015. She serves a Guest Editor of IEEE Wireless Communications, IEICE Transactions on Information and Systems and serves Editor of Peer-to-Peer Networking and Applications (Springer), Ad Hoc & Sensor Wireless Networks, International Journal of Embedded Systems (Inderscience). Her research interests include wireless sensor networks, vehicular ad hoc networks, and ubiquitous computing.

Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. Before join Muroran-IT, he was a Researcher with National Institute of Information and Communications Technology (NICT), Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong

was selected as Foreigner Research Fellow (A total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. Dr. Dongs research interests include Wireless Networks, Big Data and Cloud Computing. His research results have been published in 90 research papers in international journals, conferences and books. He is the Best Paper Award winner of IEEE HPCC 2008, IEEE ICESS 2008, ICA3PP 2014 and GPC 2015. Dr. Dong is an editor of Cyber-Physical Systems (Taylor & Francis), leading guest editor of Peer-to-Peer Networking and Applications (Springer), Journal of Multiple-Valued Logic and Soft Computing, guest editor of IEICE Transactions on Information and Systems, Mobile Information Systems, International Journal of Distributed Sensor Networks. He servers the Executive Chair of IEEE ScalCom 2014, Program Chair of IEEE SpaIoT 2014, 2015, UFirst 2014 and Program Vice Chair of IEEE CSE 2014, 2015. Dr. Dong is currently a research scientist with A3 Foresight Program (2011-2016) funded by Japan Society for the Promotion of Sciences (JSPS), NSFC of China, and NRF of Korea.

Feng Pan received M.S. degree in Industrial Automation and Ph.D. degree in Fermentation Engineering from Jiangnan University, China. He is currently a professor and Ph.D. candidate supervisor with School of Internet of Things, Jiangnan University, China. His research interests include process control, biochemical process intelligence control, computer distribute control system and Internet of Things.