Heuristic mobile data gathering for wireless sensor networks via trajectory control

Jianxin Ma¹, Shuo Shi¹, Xuemai Gu¹ and Fanggang Wang²

Abstract
This article focuses on the problem of scheduling the optimal paths of multiple mobile elements (e.g. robots, vehicles, etc.) to minimize the travel distance and balance the energy consumption and the data gathering latency in wireless sensor networks for smart cities. To partition the network for the multiple mobile elements and compute the trajectories of the multiple mobile elements, we utilize the sensor’s communication range and construct a multiple mobile elements scheduling problem. A heuristic mobile data gathering approach is proposed to solve this problem, which includes the following three steps. The sensor nodes are preliminarily partitioned into four levels, and then the clusterheads are further partitioned, and the traveling tour is scheduled for each cluster. After the first two steps, all the sensor nodes are partitioned reasonably for the multiple mobile elements. In the last step, the traveling tour is scheduled for each cluster, and the meeting point of each clusterhead is determined. We compare the proposed heuristic mobile data gathering with the existing approaches. The results indicate that the travel distance and the data gathering latency are reduced significantly, which further validates that the communication range is beneficial to minimize the travel distance.

Keywords
Data gathering, mobile elements, multiple traveling salesman problem, wireless sensor networks

Date received: 17 May 2019; accepted: 22 January 2020

Handling Editor: Michel Kadoch

Introduction
Smart city aims to improve the city’s sustainability and efficiency and ensures citizens’ quality of life and health.¹ Wireless sensor networks (WSNs) are composed of a few to hundreds or thousands of miniature battery-powered sensor nodes.²-⁴ As sensor nodes have the characteristics of low cost and small size, to realize smart city, sensors can monitor and record the physical conditions of the environment and have the ability for data processing and wireless communication.⁵

In some cases, WSNs need to transmit data to a distant central station. However, the energy of a single sensor node is not enough to communicate directly with the distant station.⁶,⁷ Based on the previous research, one of the effective methods for solving this problem is called collaborative beamforming.⁸,⁹ It is established by a set of collaborating sensors. These sensors simultaneously transmit properly weighted versions of common data such that their radiated energies are constructively combined in the direction of the central station. However, for long-distance transmission,
hundreds of sensor nodes need to transmit the common data at the same time. Mobile data gathering is another effective method. Mobile elements move in a space over time to collect data from sensor nodes. In this way, the data of a sensor node do not need to be shared with the hundreds of sensor nodes, and the energy consumption can be reduced. Moreover, as the wireless charging technology development matures, these mobile elements can collect the data from sensors and charge these sensors. The lifetime of a WSN can be extended to infinitely long for perpetual operations. Thus, we choose the second method to collect the data in this article.

Based on the survey of Yu et al., the mobility management schemes of mobile elements are divided into four categories: uncontrollable mobility (UMM), path-restricted mobility (PRM), location-restricted mobility (LRM), and unrestricted mobility (URM). These have been widely studied in recent years. For example, a hierarchical mobile element discovery protocol is proposed in Restuccia et al., which minimizes the energy consumption of sensor nodes. A novel swarm-intelligence-based sensor selection algorithm is presented to optimize network lifetime with pre-defined QoS constraints in Restuccia et al. In Ding and colleagues, a robust advantaged node placement strategy is provided to establish the robust connectivity of a graph of multiple clusters of nodes. In Yun et al., a distributed algorithm is proposed for maximizing the lifetime of a WSN when there is a mobile element, which is suitable for the delay tolerant mobile element model. According to these research, LRM that is characterized by the constraints on the mobile element location is most widely adopted by WSNs and applied in environmental monitoring. For LRM, the mobile element can only stop at certain locations for data gathering. Thus, the trajectory design and data gathering latency become significant issues.

In the large-scale WSNs, due to the data gathering latency of the mobile element and the limited capacity of the sensor node, the task scheduling for mobile elements plays a critical role in achieving a high charging and information collecting efficiency, and one mobile element is not enough. Thus, these sensor nodes are needed to assign to several mobile elements. The multiple traveling salesman problem (mTSP) is a scheduling problem, which is a generalization of the traveling salesman problem (TSP), and which aims to determine a set of routes for salesmen who start from and return to a home. In this article, based on mTSP, multiple mobile elements scheduling (mMES) problem is formulated to address these issues, which focuses on two crucial topics in this field: how to effectively and reasonably partition the network and how to choose the meeting points to obtain the optimal trajectories. It considers the situation in which the mobile elements have a single depot station. In general, mTSP can be formulated as an integer programming problem. By incorporating the communication range of each sensor node, the optimization problem becomes a non-convex mixed integer non-linear problem (MINLP) that is non-deterministic polynomial-time hard (NP-hard). If the objective and constraint functions in MINLP are convex, some exact algorithms are available to solve MINLP. However, for non-convex MINLP, it is much more difficult to be solved directly through a general solver to obtain a feasible solution. There are some specific algorithms for general non-convex MINLP. One of the typical methods is branch-and-bound which solves the problem in a tree structure. In addition, branch-and-reduce is a major step forward in the exact solution of non-convex MINLP, which has been introduced by literature. However, they are difficult for a large number of variables. In recent years, some researchers use the relaxation algorithm to obtain the optimal solution of non-convex MINLP. In Zhou et al., a new dynamic strategy is proposed for activating and deactivating mixed-integer programming (MIP) relaxations in various stages of a branch-and-bound algorithm. The authors in Zhang et al. focus on the unmanned aerial vehicle path-planning problems and propose a lossless convexification method to obtain the optimal solution by transforming the non-convex MINLP into convex programming problem.

For general mTSP, the approximation algorithm and the transformation algorithm are also used for solving it. In Kim et al., the problem of computing the optimal trajectories of multiple mobile elements is investigated to minimize data collection latency in WSNs. They assumed that any pair of neighborhoods overlap with each other and defined two problems, the k-traveling salesman problem with neighborhood (k-TSPN) and the k-rooted path cover problem with neighborhood (k-PCPN). For these two problems, they propose a constant factor approximation algorithm to solve them. In Cheng and Yu, the authors propose a data gathering approach that shortens the length of traveling path by reducing the number of visiting points and uses an approximation algorithm to plan the near-optimal traveling path. In literature, the mTSP is transformed into TSP by clustering algorithm. In Wang et al., considering the factors of the vehicles’ moving energy consumption and the limited recharging capacity, sensors are organized into clusters for easy data collection in wireless rechargeable sensor networks (WRSN). In Fu et al., novel energy synchronized mobile charging (ESync) protocol is proposed, which simultaneously reduces the charger travel distance and the charging delay. A mobile sink-based adaptive immune energy-efficient clustering protocol (MSEEIP) is proposed in Abo-Zahhad et al. to alleviate the
energy holes for data gathering in WSNs. In Abuarqoub et al.,\textsuperscript{35} a self-organizing and adaptive dynamic clustering mobile data collector (DCMDC) solution is proposed to maintain mobile data collector (MDC) relay networks, which is based on the network partitioning. The authors in Cheng et al.\textsuperscript{36} propose a seamless streaming data delivery (SSDD) protocol for multihop cluster-based WSNs with mobile elements. In Zhang et al.,\textsuperscript{37} a node-density-based clustering and mobile collection (NDCMC) approach are presented to combine the hierarchical routing and the data collection in WSNs. However, a direct solution to mMES is difficult to obtain. Based on these research, we design a heuristic mobile data gathering (HMDG) approach.

In this article, we utilize multiple mobile elements to implement the long-distance transmission for WSNs. The multiple sensor nodes partitioning methods are proposed to reduce the number of meeting points and complete the sensor nodes assignment. Based on the communication range of sensor nodes, the traveling trajectories and the meeting locations of the mobile elements are optimized. The main contributions of this article can be summarized as follows:

- A multiple mobile elements scheduling optimization problem is constructed to minimize the travel distance in WSNs. Through the use of multiple mobile elements, the long-distance data transmission of sensor nodes is implemented. By balancing the energy consumption and the data gathering latency, the energy-hole is eliminated.
- The HMDG approach is proposed for the scheduling problem of multiple mobile elements in WSNs, which has three steps. In the first step, we design a hierarchical algorithm to divide the sensor nodes into four levels, which prevents the mobile elements from stopping at every sensor node and reduces the number of meeting points. Thus, the traveling tour length and the data gathering latency can be shortened. In the second step, we provide two clustering algorithms to further partition the sensor nodes. Each sensor node will be assigned to a cluster and each cluster is assigned only one mobile element. In the last step, the optimal paths and the meeting locations are obtained by considering the communication range of sensor nodes.
- The performance of HMDG is evaluated by simulations, and the results demonstrate that HMDG can reduce the travel distance and the data gathering latency.

The remainder of this article is organized as follows. Section “Multiple mobile elements scheduling problem” formulates the mMES problem. The major contributions are introduced in section “HMDG approach,” which proposes a three-step HMDG approach to solve the mMES problem. Section “Performance evaluation” provides a particular case of our approach and compares the results of our approach with some existing approaches. Finally, section “Conclusion” concludes this article.

### Multiple mobile elements scheduling problem

#### Network model and assumption

We assume the case where \(N\) sensor nodes are randomly and relatively evenly deployed in a square region \(\mathcal{R}\) with the sides of length \(\xi\). These sensor nodes are immobile and can be recharged. Each sensor node periodically generates data packets at a generation rate. The locations of them are known \textit{a priori}. Each of them has a fixed communication range that is a circular area with radius \(r\) and unique identification (ID). The initial energy of a sensor node is \(E_0\).

There are \(M\) mobile elements out of the region \(\mathcal{R}\). These mobile elements collect data from the sensor nodes and charge the sensor nodes. The mobile elements depart from a fixed location and serve the sensor nodes in the cyclic fashion at a constant speed. The path demonstration of a mobile element has been shown in Figure 1.

#### Problem formulation

To avoid energy-hole, the traveling trajectory needs to be well scheduled. For a large-scale network, a single tour is not enough for recharging and information transmission. It costs more time, and power outage could occur. Network partitioning is an effective method to solve this problem.\textsuperscript{24} After partitioning the network, the mobile elements can visit different clusters simultaneously. We can control the number of mobile elements or the number of sensor nodes within a cluster and optimize the traveling trajectory to avoid the aforementioned problems.

Therefore, we consider the problem of scheduling the optimal trajectories of mobile elements to minimize

![Figure 1. Path demonstration of a mobile element.](image-url)
the total travel distance and find the optimal meeting locations. In applications, each sensor node has its own communication range. To find the optimal meeting locations, the mobile elements do not have to stop at the location of the sensor node. They can stop anywhere in the sensors’ communication range. Thus, the total travel distance can be saved up to $2rN$.

Regarding these issues, giving a set of mobile elements $\{a, a \in 1, 2, \ldots, M\}$ and a set of sensor nodes $\{i, i \in 1, 2, \ldots, N\}$, we formulate the mMES optimization problem as follows. Considering a graph $G = (V, E), V_i (i \in 1, 2, \ldots, N)$ is the location of sensor node $i$, which is represented by its coordinate $(v_x, v_y)$ in the two-dimensional space. $V_0$ is the sink point and the initial location of the overall mobile elements. It is represented by its coordinate $(v_{x0}, v_{y0})$ in the two-dimensional space. $(v_{px}, v_{py})$ is the optimized meeting location within the communication range of node $i$. $E$ is the set of edges, and the number of all elements in $E$ is $N(N + 1)/2$. Each edge $E_{ij}$ is associated with a travel distance $c_{ij}$. Note that $c_{0j}$ is the distance from the initial point $V_0$ to the first sensor node to be visited, and $c_{0j}$ is the distance from the last sensor nodes to be visited to the initial point. $G_b$ is the set of the sensor nodes in the $b$th cluster.

$$\begin{align*}
\text{minimize} & \quad \sum_{b=1}^{M} \left( \sum_{i \in G_b} c_{0i}x_{0i} + \sum_{b \in G_b} \sum_{i \in G_b} c_{ij}x_{ij} + \sum_{b \in G_b} c_{0b}x_{0b} \right) \\
\text{subject to} & \quad 0 \leq \theta \leq 2\pi \\
& \quad x \in \{0, 1\} \\
& \quad \sum_{i \in G_b} x_{0i} = \sum_{i \in G_b} x_{0b} = 1 \\
& \quad \sum_{i \in G_b} x_{ij} = N_{G_b} - 1 \\
& \quad L_b < T_b \\
& \quad \mu_j^{(b)} \geq \mu_j^{(b)} + 1 - (N_{G_b} - 1)(1 - x_{ij}) 
\end{align*}$$

where $c_{ij} = \sqrt{v_{px} - v_{xj}}^2 + v_{py} - v_{yj}^2$, $v_{xj} = v_{xj} + r \sin \alpha_j$, $v_{yj} = v_{yj} + r \cos \alpha_j$, $v_{px} = v_{px} + r \sin \beta_j$, and $v_{py} = v_{py} + r \cos \beta_j$.

Equation (1b) is the bound constraint for angle $\theta$, and $\theta$ includes $\alpha_i$ and $\beta_j$. Equation (1c) imposes $x$ that includes $x_{ij}, x_{0i}$, and $x_{0j}$ to be 0 or 1 valued. If a mobile element visits node $j$ from node $i$, the decision variable $x_{ij}$ is 1, otherwise it is 0. Equations (1d) and (1e) ensure that the path is connected and each sensor node is visited once only. $N_{G_b}$ is the number of the sensor nodes in the $b$th cluster. Equation (1f) guarantees the data gathering latency $L_b$ shorter than the minimum sensor node lifetime $T_b$ in the $b$th cluster. That is, the mobile element will arrive at the sensor node before the sensor node is no power. As only one mobile element will be assigned to visit all the sensor nodes within a cluster, we do not want more than one tour as the scheduling result for each cluster. The redundant tour is named as sub-tour. $\mu_j^{(b)}$ is the number of the stops along the tour at which the sensor node $i$ is visited in the $b$th cluster. Equation (1g) eliminates the sub-tour within a cluster, which is formulated from Gavish.$^{39}$

In this problem, the objective function includes the integer variables $x_{ij}, x_{0i}$, and $x_{0j}$ and non-integer variables $\alpha_i$ and $\beta_j$. Thus, this is a mixed integer problem. For a mixed integer problem, its feasible search space exists a set of discrete points because of the integer variables. After relaxing the integer variables, based on the gradient and the Hessian matrix of the objective function, the objective function is a non-convex MINLP. The constraints include bound constraints, binary variables, and linear equalities. Generally, it is difficult to be solved directly. Therefore, we propose the HMDG to solve the above optimization problem.

### HMDG approach

From inter-cluster to intra-cluster, HMDG solves the mMES optimization problem by three steps: (1) Based on the number of neighbors of each sensor node, the four-level network is built. (2) Two clustering algorithms are presented to partition the clusterheads of the four-level network. (3) The optimized traveling trajectories and meeting locations are scheduled. The flowchart of HMDG is shown in Figure 2.

**Inter-cluster: network partitioning**

First of all, the mobile elements know the serial numbers and the location information of sensor nodes. The initial level setting is $N_{level} = -1$ for each sensor node, which indicates that the sensor node $i$ has not

```
Randomly deploy N sensor nodes in a square region.
Step 1: Find neighbors.
Low

M/N

High

Step 2: Distance-sensitive clustering algorithm.
Step 3: Schedule the traveling trajectory of each cluster.
End
```

**Figure 2.** Flowchart of HMDG.
Algorithm 1. Preliminarily partitioning the sensor nodes

1: Initialization: For all sensor nodes, the initial setting is 
\( N_{\text{level}} = -1, N_{\text{head}} = 0; \)
2: Calculate the distance between any two sensor nodes;
3: Count the number of neighbors \( N_b \) of each sensor node within its 1/2 communication radius;
4: for \( i = 1, i \leq N_b + 1 \) do
5: if \( N_b \geq TH \) then
6: Sensor node \( i \) becomes a clusterhead
7: \( N_{\text{level}} = 0, N_{\text{head}} = 1 \)
8: Its neighbors become its childnodes
9: For the childnode \( j, N_{\text{level}} = 1, N_{\text{parent}} = i; \)
10: end if
11: if \( 1 \leq N_b < TH \) and \( N_{\text{level}} = -1 \) then
12: Sensor node \( i \) becomes a clusterhead
13: \( N_{\text{level}} = 2, N_{\text{head}} = 1 \)
14: Its neighbors become its childnodes
15: For the childnode \( j, N_{\text{level}} = 1, N_{\text{parent}} = i; \)
16: end if
17: if \( N_b = 0 \) and \( N_{\text{level}} = -1 \) then
18: Sensor node \( i \) becomes a clusterhead
19: \( N_{\text{level}} = 3, N_{\text{head}} = 1; \)
20: end if
21: end for

joined any of the clusters yet. After the initial step, each sensor node has a set of neighbors within the circle of its 1/2 communication radius. As shown in Algorithm 1, we explain how to find the neighbors of each sensor node and how a sensor node becomes a clusterhead or a child node.

For the first step of HMDG, each sensor node needs to estimate whether the number of its neighbors is less than the threshold value \((TH = \lceil N/M \rceil)\), \( \lceil x \rceil \) returns the first integer that is larger than \( x \). If the number of neighbors of a sensor node is greater than or equal to \( TH \), the sensor node becomes a clusterhead \((N_{\text{level}} = 0)\) and its neighbors become its child nodes \((N_{\text{level}} = 1)\). If the number of neighbors of a sensor node is less than \( TH \) and greater than or equal to 1, the sensor node becomes a clusterhead \((N_{\text{level}} = 2)\) and its neighbors become its child nodes \((N_{\text{level}} = 1)\). If a sensor node has no neighbors, it becomes the clusterhead \((N_{\text{level}} = 3)\) of itself.

After preliminarily partitioning, we obtain a four-level network. The sum of the number of sensor nodes of these levels is \( N \), which can be represented as \( N = N_{\text{level} 0} + N_{\text{level} 1} + N_{\text{level} 2} + N_{\text{level} 3} \). Based on the four-level network, in the second step, the sensor nodes will be further partitioned. The cluster that has more than \( TH \) sensor nodes can be directly assigned to a mobile element in the second step. It composes a super-cluster by itself. For the clusters that do not have enough sensor nodes to build the super-cluster, a clustering algorithm will be used to partition them again. We present two such algorithms, and they will be compared in the following section.

**Distance-sensitive clustering algorithm.** In the network, the distance that is from the central station to the cluster may be sensitive for each cluster. Thus, how to control the distance between the central station and the cluster is an important research problem. Based on this issue, we design a distance-sensitive clustering algorithm to shorten this portion of the travel distance.

In order to satisfy this requirement, we need to choose some meeting points that are close to the central station as short as possible. Furthermore, all the sensor nodes whose levels are \( N_{\text{level}} = 2 \) and \( N_{\text{level}} = 3 \) need to be joined this process.

First, we divide the edge that is near the central station of the square region into \( M - N_{\text{level} 0} \) sub-regions, as shown in Figure 3. As all the sensor nodes are relatively evenly deployed in the square region, no sub-region has too many sensor nodes. If there is a super-cluster through multiple sub-regions, some sub-regions may be merged. This depends on the number of sensor nodes within these sub-regions.

Therefore, we have the clusters \( G_{d,1}, G_{d,2}, \ldots, G_{d, M - N_{\text{level} 0}} \).

**K-means clustering algorithm.** We also can use the well-known K-means clustering algorithm\(^{40}\) to further partition these sensor nodes \((N_{\text{level}} = 2)\) and \( N_{\text{level}} = 3 \), as shown in Figure 4.

Using the K-means clustering algorithm, the sensor nodes are assigned to the specific cluster based on the minimized Euclidean distance regarding the centroid of each region. It aims to minimize the following objective function

\[
S = \sum_{b = 1}^{M - N_{\text{level} 0}} \sum_{i \in G_b} \left\| x_i^{(b)} - \delta_i^{(b)} \right\|^2
\]  

\[(2)\]

![Figure 3. Demonstration of distance-sensitive clustering in the second step.](image-url)
in which \( \| \cdot \| \) is the Euclidean distance between a sensor node \( i \) of the cluster \( b \) and the region’s centroid \( \delta^{(b)} \). \( V_i^{(b)} \) denotes the coordinate location \((v_{x_i}, v_{y_i})\) of the sensor node \( i \), and the sensor node has been assigned to the \( b \)th cluster.

Initially, we randomly select \( M - N_{\text{level}_b} \) sensor nodes from \( N_{\text{level}_b} = 2 \) and \( N_{\text{level}_b} = 3 \) sensor nodes as the initial centroid, which are uniformly distributed in the network. Then, every sensor node will be assigned to the closest centroid. After all the sensor nodes have been associated with a centroid, the new distance will be calculated to find the new centroids in the new clusters.

This process is repeated until the centroids do not change any more. Therefore, we have the clusters \( G_k,1, G_k,2, \ldots, G_k,j, \ldots, G_k,M - N_{\text{level}_b} \).

**Intra-cluster: scheduling the traveling tour for each cluster**

In the third step, we focus on how to schedule the optimal paths and meeting locations of the mobile elements. For these selected sensor nodes whose levels are \( N_{\text{level}_b} = 0, N_{\text{level}_b} = 2 \), and \( N_{\text{level}_b} = 3 \), the mobile elements will stop at the circle of their 1/2 communication radius.

We adopt the two-step solver\(^{41} \) to solve this non-convex MINLP problem. After running the solver, each cluster will obtain an optimized path and the location information of the meeting points. In this step, in order to reduce the energy consumption of clusterheads, all the sensor nodes directly transmit data to the mobile element.

**Analysis of the network lifetime, energy consumption, and data gathering latency**

To effectively transmit data and charge sensor nodes, the mobile elements need to arrive at each sensor node before these sensor nodes have no power. Thus, we need to know the lifetime of sensor nodes.

A sensor node receives a beacon signal from a mobile element and estimates the transmission distance according to the beacon signal strength. Then, the sensor node can adjust the transmission power according to the distance to the mobile element.\(^{42} \) Thus, we consider the distance-dependent energy consumption model\(^{43} \) from the data transmission and reception.

As mentioned above, the sensor nodes directly transmit data to the mobile element. To deliver \( l \) bits of data over the distance of \( d \), the energy consumption will be

\[
E_{tx}(l, d) = E_{tx-\text{elec}}(l) + E_{tx-\text{amp}}(l, d) = lE_{\text{elec}} + lfsd^\alpha
\]

where \( E_{\text{elec}} \) is the circuits’ power consumption for processing the data. \( fs \) is the power for radio amplification to compensate for the path loss. \( \alpha \) is the path loss exponent.\(^{44} \)

For the path loss, the longest distance between a mobile element and a sensor node is \( r \). If a mobile element stops at the location of a sensor node, the shortest transmission distance will be 0. \( d_{\text{bmax}} \) denotes the longest transmission distance in the \( b \)th cluster. Therefore, the maximum power loss to deliver \( l \) bits of data to the mobile element for a sensor node in \( b \)th cluster is

\[
E_{tx}^b(l, d) = l\left(E_{\text{elec}} + lfsd_{\text{bmax}}^\alpha\right)
\]

The average power loss to deliver \( l \) bits of data to the mobile element for a sensor node is

\[
E_{tx}(l, d) = \frac{1}{r} \int_{x=0}^{r} E_{tx}(l, d) dx
\]

\[
= \frac{1}{r} \left( lE_{\text{elec}} + lfsd_{\text{max}}^\alpha \right) dx
\]

\[
= lE_{\text{elec}} + lfsd_{\text{bmax}}^\alpha / \alpha + 1
\]

Although a sensor node needs to receive the beacon signal from a mobile element, it requires much smaller energy than sending the data to the mobile element. Thus, we assume \( E_{rx} = 0 \). Then, the lifetime of the \( b \)th cluster is

\[
T_b = \frac{E_0}{E_{tx}^b(l, d) + E_{rx}}
\]

\[
= \frac{E_0}{l\left(E_{\text{elec}} + lfsd_{\text{bmax}}^\alpha\right)}
\]

The estimated network lifetime per data gathering round is
\[ T = \frac{E_0}{E_{\text{tx}} + E_{\text{rx}}} = \frac{E_0}{l(E_{\text{elec}} + efs \frac{r^a}{a + 1})} \]  

Data gathering latency mainly depends on three parts: the sum of the parking time at the meeting points, the traveling time through all the meeting points, and the parking time at the central station.\(^2^4\) For a cluster, it can be calculated as

\[ L_b = \frac{D_{G_b}}{s_{ab}} + N_{G_b} \tau + t_0 \]  

and then, for the inter-cluster, it is

\[ L = \max \{L_b\} \]  

where \(D_{G_b}\) denotes the path length of the \(b\)th cluster. \(\tau\) denotes the time duration in which the mobile element \(a\) stays for receiving the data from a sensor node and charging the sensor node. \(t_0\) is the time taken for sending the data to the central station and being charged at the central station for a mobile element. The mobile element \(a\) that is assigned to the \(b\)th cluster runs at the speed of \(s_{ab}\). Thus, minimizing the travel distance \(D_{G_b}\) is an effective method to minimize the data gathering latency \(L_b\).

### Performance evaluation

In this section, the algorithm procedure is presented by applying our HMDG to a synthetic example. Then, the performance of HMDG is evaluated by comparing them with a classic TSP-based approach and an mTSP-based approach. Distance-sensitive clustering algorithm and \(K\)-means clustering algorithm are also compared in the second step of HMDG. In the end, it is verified that all the sensor nodes can be visited by the mobile elements in each data gathering round.

### An example of HMDG

To illustrate the algorithm, we randomly generate 500 sensor nodes in the size of \(900 \times 900 \text{m}^2\). Then we use HMDG algorithm to partition the 500 sensor nodes into 20 clusters. The results are shown in Figures 5 and 6 where different levels and clusters are labeled with different colors and icons. As shown in Figure 5(a), all the sensor nodes have been divided into four levels after the first step of HMDG. The green plus sign icons represent the \(N_{i, \text{level}} = 0\) sensor nodes. The red asterisk icons identify \(N_{i, \text{level}} = 2\) sensor nodes. The blue circle icons show \(N_{i, \text{level}} = 3\) sensor nodes. The child nodes \((N_{i, \text{level}} = 1)\) are labeled with the yellow upward-pointing triangle icons. Figure 5(b) is the simulation result of using the distance-sensitive clustering algorithm in the second step of HMDG, and Figure 5(c) is the simulation result of using the \(K\)-means clustering algorithm in the second step of HMDG. For each cluster, only one mobile element will be assigned to these sensor nodes. In the third step, the traveling tour for each cluster has been scheduled. To clearly show that, we choose three of them as shown in Figure 6. All of them are the tours that come from and back to the sink point (1000, 450).

### Investigating the network lifetime

The network lifetime with different size of communication radius is investigated in this section. A total of 100 sensor nodes and 5 mobile elements are deployed in the size of \(200 \times 200 \text{m}^2\). The communication radius of sensor nodes varies from 0 to 50. We assume that the initial energy of a sensor node is \(E_0 = 0.5 \text{ J}\) and the path loss exponent is \(\alpha = 2\). We employ equations (4) and (5) to calculate the energy consumption, where \(E_{\text{elec}} = 50 \text{ nJ/bit}\) and \(ef_s = 10 \text{ PJ/bit/m}^a\). The data generation rate of a sensor node is 4 bits/s.

The simulation results of the network lifetime are shown in Figure 7. The red line represents the analytical
average results, the green line represents the simulation results, and the other lines show the minimum lifetime of each cluster. With the increase of the communication radius, the network lifetime is decreased. The simulation results match the analytical results. The minimum lifetime of each cluster depends on the distribution of the sensor nodes within this cluster. The farther the distance between the meeting point and the sensor node is, the smaller the lifetime is. If the distance is equal to $r$, the lifetime is minimum. The smaller communication radius leads to less energy consumption and the longer network lifetime, but the travel distance and the data gathering latency will be extended. Based on this, we balance the energy consumption and the data gathering latency in order that the mobile elements will arrive at each sensor node before the sensor node has no power. Thus, energy-hole can be eliminated.

**Investigating the data gathering latency**

The data gathering latencies between different methods are compared in Figure 8. A total of 100 sensor nodes are randomly deployed in the size of $200 \times 200$ m$^2$. We assume $t_0 = 900$ s and each sensor node generates a data packet at every 1 s. The movement speed of mobile elements varies from 5 to 30 m/s. The mTSP-based approach uses K-means to partition the network and the classic TSP to schedule the traveling tour of each cluster. It can be seen that increasing the movement speed of the mobile element can reduce data gathering latency. The data gathering latency of the TSP-based approach is obviously longer than the other two approaches, and HMDG achieves the best performance for all the speed in the figure. This shows HMDG is an effective method to reduce the data gathering latency.

**Investigating the travel distance**

The total travel distance of using the HMDG approach is compared with an mTSP-based approach in this section. The results when the number of sensor nodes varies from 20 to 100 are plotted in Figure 9, and there are two mobile elements. We vary the number of sensor nodes within the network to investigate the impact of the density of the network between these two methods. As shown in this figure, when we use the HMDG, the total travel distance is shorter than the case of the K-means-based mTSP approach. Meanwhile, we consider...
the impact of the number of mobile elements in Figure 10. As all the mobile elements depart from the sink point, travel in the network, and return back to the same sink point, the total travel distance includes these three parts. Thus, increasing the number of mobile elements may lead to an increase in the total travel distance, but the path length for each cluster decreases. This improves the data gathering efficiency of the large scale network.

Comparison the clustering algorithms in the second step

In section “Inter-cluster: network partitioning,” two methods as the second step of HMDG are presented to partition the clusterheads. In this part, based on the different sensing area size and the different number of mobile elements, we compare the total travel distance using these two methods. Figure 11 shows the total travel distance results when the number of sensor nodes varies from 20 to 100. As shown in this figure, using the distance-sensitive clustering algorithm as the second step of HMDG can obtain a shorter path than using the K-means clustering algorithm as the second step of HMDG when fewer mobile elements serve in the network. Thus, the distance-sensitive clustering algorithm performs better when the ratio of the number of mobile elements to the number of sensor nodes is low, and when there are not many mobile elements in the network.

Impact of the size of communication range

From the above analysis, it is shown that HMDG is an effective method to shorten the travel distance and eliminate the energy-hole. We discuss more details on the impact of the size of the communication radius on the original problem in this section.

We vary the value of communication radius from 10 to 50 m, respectively, for each fixed number of sensor nodes from 20 to 100. Figure 12 presents the results when there are five mobile elements. From this figure, we can see that the travel distance is decreased with the increasing of the size of the communication radius. Considering the communication range can save more movement energy consumption than we do not consider it. However, these sensor nodes are randomly deployed in the field. The result may have some fluctuation each time, which depends on the layout of these sensor nodes.

Investigating the visit rate

In this section, the visit rate is investigated, which verifies that all the sensor nodes can be visited in each round using HMDG. In the example, 100 sensor nodes are randomly deployed in the size of 200 × 200 m², and each of them has a unique ID from 1 to 100. There are four mobile elements which are being prepared to visit them. After running the first step of HMDG, these sensor nodes are divided into four levels, as shown in Table 1. Table 1 presents the serial number of each level, and the sum of the number of these sensor nodes is 100. For the second step, the partitioning results are shown in Table 2. As only the sensor nodes whose levels are \( N_{\text{level}} = 2 \) and \( N_{\text{level}} = 3 \) are joined this process, the sum of the number of the sensor nodes in Table 2 is less than 100. However, at each meeting point, the mobile element not only visits the sensor node in the center but also visits all the sensor nodes within this range. Thus, we calculate the number of sensor nodes in Table 2 and the number of super-clusters.
Ni level = 0 and child nodes (Ni,level = 1), which is equal to 100. Therefore, all the sensor nodes have been allocated to the mobile elements, and all of them can be visited by mobile elements.

Figure 11. Comparison results between the distance-sensitive clustering algorithm and the K-means clustering algorithm with the different size of network and the different number of mobile elements: (a) area size is 200 × 200 and the number of mobile elements is two, (b) area size is 200 × 200 and the number of mobile elements is four, (c) area size is 200 × 200 and the number of mobile elements is 10, (d) area size is 400 × 400 and the number of mobile elements is two, (e) area size is 400 × 400 and the number of mobile elements is four, (f) area size is 400 × 400 and the number of mobile elements is 10, (g) area size is 600 × 600 and the number of mobile elements is two, (h) area size is 600 × 600 and the number of mobile elements is four, and (i) area size is 600 × 600 and the number of mobile elements is 10 (area in m²).

Table 1. Simulation results of the fist step of HMDG.

| Level | Serial number | Sum   |
|-------|---------------|-------|
| 0     | 1 3 8 9 10 13 15 17 20 21 24 25 27 28 29 31 36 37 39 44 45 46 50 51 52 53 54 55 | 0     |
| 1     | 57 59 62 63 65 67 70 71 72 73 76 79 80 81 85 86 89 91 92 93 95 98 99 100 | 52    |
| 2     | 4 5 6 7 11 14 18 19 23 26 32 33 34 40 41 42 43 48 56 58 60 66 68 69 75 78 82 87 90 96 | 30    |
| 3     | 2 12 16 22 30 35 38 47 49 61 64 74 77 83 84 88 94 97 | 18    |

Conclusion

This article mainly focuses on the optimization of the traveling tour for multiple mobile elements in large-scale WSNs. Regarding the capacity limit, the energy
consumption, and the data gathering latency, we formulated an mMES optimization problem. To solve this problem, we proposed a HMDG approach that can properly partition the network for the multiple mobile elements and schedule the optimal trajectories of these mobile elements based on the optimal meeting points. We evaluated and compared the proposed approach by extensive simulations. The results showed that both the travel distance and the data gathering latency are reduced significantly. The communication range of sensor nodes is beneficial to shorten the travel distance and reduce the data gathering latency.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the project “the Verification Platform of Multi-tier Coverage Communication Network for Oceans (PCL2018KP002).”

ORCID iD
Jianxin Ma https://orcid.org/0000-0002-2518-8367

References
1. Liu J, Ke X, Fan P, et al. RF energy harvesting wireless powered sensor networks for smart cities. *IEEE Access* 2017; 5(99): 9348–9358.
2. Siew ZW, Chin YK, Kiring A, et al. Comparative study of various cluster formation algorithms in wireless sensor networks. In: *Proceedings of the 7th international conference on computing and convergence technology (ICCCT)*, Seoul, South Korea, 3–5 December 2012, pp. 772–777. New York: IEEE.
3. Watteyne T, Molinaro A, Richichi MG, et al. From MANET to IETF ROLL standardization: a paradigm shift in WSN routing protocols. *IEEE Commun Surv Tutor* 2011; 13(4): 688–707.
4. Shuai H, Gong Z, Meng W, et al. Automatic precision control positioning for wireless sensor network. *IEEE Sens J* 2016; 16(7): 2140–2150.
5. Boyinbode O, Le H and Takizawa M. A survey on clustering algorithms for wireless sensor networks. *Int J Space Comput* 2011; 12–3: 130–136.
6. Perillo M and Heinzelman W. An integrated approach to sensor role selection. *IEEE Trans Mobile Comput* 2009; 8(5): 709–720.
7. Van Le D, Oh H and Yoon S. HiCoDG: A hierarchical data-gathering scheme using cooperative multiple mobile elements. *Sensors* 2014; 14(12): 278–304.
8. Haro BB, Zazo S and Palomar DP. Energy efficient collaborative beamforming in wireless sensor networks. *IEEE T Signal Pr* 2014; 62(2): 496–510.
9. Salari S, Kim IM and Chan F. Distributed cooperative localization for mobile wireless sensor networks. *IEEE Wirel Commun Lett* 2017; 99: 1–1.
10. Tunca C, Işik S, Donmez MY, et al. Distributed mobile sink routing for wireless sensor networks: a survey. *IEEE Commun Surv Tutor* 2014; 16(2): 877–897.
11. Yu S, Zhang B, Li C, et al. Routing protocols for wireless sensor networks with mobile sinks: a survey. *Commun Mag IEEE* 2014; 52(7): 150–157.
12. Zhu C, Shu L, Hara T, et al. A survey on communication and data management issues in mobile sensor networks. *Wirel Commun Mobile Comput* 2014; 14(1): 19–36.
13. Fu L, Cheng P, Gu Y, et al. Optimal charging in wireless rechargeable sensor networks. *IEEE T Veh Technol* 2016; 65(1): 278–291.
14. Pozza R, Nati M, Georgoulas S, et al. Neighbor discovery for opportunistic networking in internet of things scenarios: a survey. *IEEE Access* 2015; 3(3): 1101–1131.
15. Yu G, Ren F, Ji Y, et al. The evolution of sink mobility management in wireless sensor networks: a survey. *IEEE Commun Surv Tutor* 2017; 18(1): 507–524.
16. Restuccia F, Anastasi G, Conti M, et al. Analysis and optimization of a protocol for mobile element discovery in sensor networks. *IEEE Trans Mob Comput* 2014; 13(9): 1942–1954.
17. Restuccia F and Das SK. Lifetime optimization with QoS of sensor networks with uncontrollable mobile
18. Ding K, Yousefi'zadeh H and Jabbari F. A robust advantaged node placement strategy for sparse network graphs. *IEEE T New Sci Eng* 2018; 5(2): 113–126.
19. Ding K and Yousefi'zadeh H. A systematic node placement strategy for multi-tier heterogeneous network graphs. In: *Proceedings of the IEEE wireless communications and networking conference (WCNC)*, Doha, Qatar, 3- April 2016, pp. 1–6. New York: IEEE.
20. Yun YS, Xia Y, Behdani B, et al. Distributed algorithm for lifetime maximization in delay-tolerant wireless sensor network with mobile sink. *IEEE T Mobile Comput* 2013; 12(10): 1920–1930.
21. Zhou Z, Du C, Shu L, et al. An energy-balanced heuristic for mobile sink scheduling in hybrid WSNs. *IEEE T Ind Inform* 2016; 12(1): 28–40.
22. Bektas T. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* 2006; 34(3): 209–219.
23. Sundar K and Rathinam S. Generalized multiple depot traveling salesmen problem—polyhedral study and exact algorithm. *Comput Oper Res* 2016; 70: 39–55.
24. Wang C, Li J, Ye F, et al. A mobile data gathering framework for wireless rechargeable sensor networks with vehicle movement costs and capacity constraints. *IEEE T Comput* 2016; 65(8): 2411–2427.
25. Gu X, Zhou X, Yuan B, et al. A bayesian compressive data gathering scheme in wireless sensor networks with one mobile sink. *IEEE Access* 2018; 6: 897–910.
26. Burer S and Letchford AN. Non-convex mixed-integer nonlinear programming: a survey. *Surv Oper Res Manag Sci* 2012; 17(20): 97–106.
27. McCormick GP. Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Math Program* 1976; 10(1): 147–175.
28. Hong SR and Sahinidis NV. A branch-and-reduce approach to global optimization. *J Global Optim* 1996; 8(2): 107–138.
29. Tawarmalani M, Sahinidis NV and Sahinidis N. Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications, vol. 65. New York: Springer, 2002.
30. Zhou K, Kilnç MR, Chen X, et al. An efficient strategy for the activation of MIP relaxations in a multicore global MINLP solver. *J Global Optim* 2018; 70(3): 497–516.
31. Zhang Z, Wang J and Li J. Lossless convexification of nonconvex MINLP on the UAV path-planning problem. *Optimal Control Appl Method* 2017; 39(1): 845–859.
32. Kim D, Abay BH, Uma R, et al. Minimizing data collection latency in wireless sensor network with multiple mobile elements. In: *Proceedings of the IEEE INFOCOM*, Orlando, FL, 25–30 March 2012, pp. 504–512. New York: IEEE.
33. Cheng CF and Yu CF. Data gathering in wireless sensor networks: a combine–TSP–reduce approach. *IEEE T Veh Technol* 2016; 65(4): 2309–2324.
34. Abo-Zahhad M, Ahmed SM, Sabor N, et al. Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks. *IEEE Sens J* 2015; 15(8): 4576–4586.
35. Abuqroub A, Hammoudeh M, Adebisi B, et al. Dynamic clustering and management of mobile wireless sensor networks. *Comput Netw* 2017; 117: 62–75.
36. Cheng L, Niu J, Francesco MD, et al. Seamless streaming data delivery in cluster-based wireless sensor networks with mobile elements. *IEEE Syst J* 2016; 10(2): 805–816.
37. Zhang R, Pan J, Xie D, et al. NDCMC: A hybrid data collection approach for large-scale WSNs using mobile element and hierarchical clustering. *IEEE Internet Things J* 2016; 3(4): 533–543.
38. Fu L, He L, Cheng P, et al. ESync: an energy synchronized charging protocol for rechargeable wireless sensor networks. *IEEE T Veh Technol* 2016; 65(9): 7415–7431.
39. Gavish B. Note: a note on “the formulation of the m-salesman traveling salesman problem.” *Manag Sci* 1976; 22(6): 704–705.
40. MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Berkeley, CA, June 21–July 18, 1965. University of California press, 1967.
41. Ma J, Shi S and Gu X. An optimization-based mTSP-CR mobile data gathering algorithm for large-scale wireless sensor networks. In: *Proceedings of the IEEE 88th vehicular technology conference (VTC-Fall)*, Chicago, IL, 27–30 August 2018, pp.1–5. New York: IEEE.
42. Liu J, Jiang X, Nishiyama H, et al. Throughput capacity of MANETs with power control and packet redundancy. *IEEE T Wirel Commun* 2013; 12(6): 3035–3047.
43. Wang AY and Sodini C.G. A simple energy model for wireless microsensor transceivers. In: *Proceedings of the IEEE global telecommunications conference, GLOBECOM’04*, vol. 5, Dallas, TX, 29 November–3 December 2004, pp.3205–3209. New York: IEEE.
44. Heinzelman WB, Chandrakasan AP and Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. *IEEE T Wirel Commun* 2002; 1(4): 660–670.