Abstract

Adopting mobile data gathering in wireless sensor networks (WSNs) can reduce the energy consumption on data forwarding thus achieve more uniform energy consumption among sensor nodes. However, the data delivery latency inevitable increases in mobile data gathering due to the travel of the mobile sink. In this paper, we consider a delivery latency minimization problem (DLMP) in a randomly deployed WSN. Our goal is to minimize the travel latency of the mobile sink. We formulate the DLMP as an integer programming problem which subjects to the direct access constraint, the data transmission constraint and the route traverse constraint. We prove that the DLMP is an NP-Complete (NPC) problem, and then propose a substitution heuristic algorithm to solve it by shortening the travel route and having the mobile sink move and collect data at the same time. We compare the proposed algorithm with other two algorithms, a traveling salesman problem (TSP) heuristic algorithm and a random heuristic algorithm through simulations.
Our extensive simulation results show that although all the three algorithms can shorten the data delivery latency in mobile data gathering, the proposed substitution heuristic algorithm is the most effective one.

**Keywords:** Wireless sensor networks, mobile sink, delivery latency minimization, substitution heuristic algorithm.

### 1. Introduction

A wireless sensor network with mobile sink (WSN-MS) provides an effective method to collect data than the wireless sensor networks (WSNs) with static sink. In the traditional WSNs with static sink, the relaying forward often causes hot spot problem [1, 2], that is, sensor nodes close to the sinks deplete their energy quickly. On the contrary, in a WSN-MS, although sensor nodes are still stationary, mobile sinks can access sensor nodes by moving around them thus data from sensor nodes can be transmitted to the sinks directly or through less relaying to save more energy and achieve longer lifetime. Moreover, mobile sinks can also access the disconnected WSNs. However, WSNs-MS still have a crucial problem that the delivery latency may be long due to the traveling of mobile sinks [3].

In a WSN-MS, the delivery latency is defined as the time mobile sinks take to traverse the sensing range of sensor nodes once to collect data and deliver the collected data to the base station [1],[4], [5]. The main reason which causes long delivery latency is the long moving time of mobile sinks. There are many challenges to minimize the delivery latency in WSNs-MS. First, the number of candidate turning positions for mobile sinks is infinite, and it is difficult to determine which one is the best. In [4], the selection of polling points is proved to be an NP-hard problem, and the polling points are selected by achieving the maximum compatible pairs among sensor nodes. For simplicity, most of other works [6],[7],[8],[9],[10] make assumption that the candidate positions are given. Second, the travel route is hard to plan for the circular travel route of the mobile sink which is formulated as a traveling salesman problem (TSP). This problem is also an NP-Complete (NPC) problem. For simplicity, some works treat the movement of mobile sinks as random walk, such as [11]. Other works use predefined trajectory, such as [12], [13], [14], and even some works simply ignore the moving time and moving path planning, such as [7], [15].

In this paper, we aim to minimize the data delivery latency in a WSN-MS composed of a mobile sink and some stationary sensor nodes which are randomly deployed on a plane. We assume that the positions of both the mobile sink and sen-
Sensor nodes are known in advance through the equipped global positioning system (GPS) [16] on the mobile sink, location service [17], or special guiding mechanisms [18], [19]. The mobile sink visits sensor nodes directly; in other words, the mobile sink must lie in the communication range of sensor nodes to collect data. Our main idea is that the mobile sink traverses sensor nodes in the network and simultaneously collects their data to minimize the delivery latency. One data collection cycle is defined as that the mobile sink departs from the origin position, traverses every sensor node and gets back to the origin position.

The contribution of this work can be summarized as follows. First, we formulate the delivery latency minimization problem (DLMP) as an integer programming. Second, we further prove that the DLMP is an NPC problem, show that the anchor points should be located at the border of communication range of sensor nodes such that the length of travel route would be short, and prove that the lower bound of delivery latency is the sum of transmission time of sensor nodes. Third, we propose a substitution heuristic algorithm to plan the travel route of the mobile sink by point substitution and line substitution and minimize the delivery latency by a relaxed linear programming. Finally, our extensive simulations validate the effectiveness of the proposed heuristic algorithm in terms of shortening route length and reducing delivery latency by comparing with a TSP heuristic algorithm and a random heuristic algorithm. In the simulation, we further study the effects of moving speed of the mobile sink, transmission time and communication radius of sensor nodes on algorithm performance.

The rest of the paper is organized as follows. Section 2 summarizes the related works. Section 3 introduces the system model and formulates the DLMP. Section 4 proposes the substitution heuristic algorithm. Section 5 presents the simulation results and some discussions. Finally, Section 6 concludes the paper.

2. Related Works

In recent years, several implementation techniques and network architectures of WSNs-MS have been proposed, which show that WSNs-MS are an effective approach for data gathering. In [20], Kansal et al. implemented a prototype system of WSNs-MS which is composed of a mobile router and several static motes. In [21], Zhao and Yang proposed a three-layer framework to achieve good scalability, long network lifetime and low data collection latency. In [14], Vlajic and Stevanovic simulated that the idealistic (zero-overhead) WSNs-MS can distribute routing load and prolong network lifetime. In [22], Pazzi et al. proposed a data dissemination protocol for a WSN with mobile sinks which combines cluster for-
mation, sleep-wake mechanism, trail generation and path recovery. The above works provide the foundation for WSN-MS.

In addition, several studies show that the travel tour of mobile sinks greatly affects the network efficiency. In [16], Xu et al. locate the sensor nodes by equipping GPS. In [18], [17], [19], the authors propose approaches to locate the sensor nodes and mobile sink. In [11], Shah et al. utilize Data MULEs to collect data generated by randomly distributed wireless sensor nodes, and transfer it to access points. In this scheme, Data MULEs is a type of serendipitous mobile agents whose movements cannot be predicted in advance. In [13], Chakrabarti et al. propose a predictable observer mobility model and an observer-driven communication protocol in reducing energy consumption for data collection, which could be regarded as the second type of mobility. In [3], [4], [21] and [23], the moving tour of mobile sink is controllable, so that the metrics such as the network time and the energy consumption of sensor nodes which rely on the moving tour can be optimized. In [3], Ma and Yang utilize a mobile data collector, SenCar, to periodically traverse sensor nodes and collect data by clustering sensor nodes via multi-hop routing, and find that the moving path of SenCar greatly affects network lifetime. In [23], they further propose a data gathering algorithm to minimize the length of each data gathering tour with multiple collectors.

In the meanwhile, minimizing data gathering time in WSNs-MS has also been studied. Zhao et al. [4] adopt the mobility and space-division multiple access technique to minimize the total data gathering time in a WSN with a single or multiple mobile sinks, which is mostly related to our work in this paper. In [4], multiple antennas are equipped on each the mobile sink so that distinct compatible sensor nodes may upload data concurrently, and the data gathering time problem is formulated as a problem of finding compatible pairs among sensor nodes, determining sensor association pattern and finding locations of selected polling points and the order for the mobile sink to visit. In contrast, in this paper we consider the scenario that the mobile sink traverses sensor nodes and at the same time gathers their data in a ubiquitous WSN-MS in which the mobile sink is only equipped one antenna. We further define the data gathering time as the delivery latency, and study the effects of moving speed, data transmission time and communication radius of sensor nodes.
3. System Model and Problem Statement

3.1. System Model

In this paper, we consider a WSN with a single mobile sink. We assume that (1) sensor nodes are deployed on a plane randomly, and their positions can be measured by location technology [16], [17]; (2) the communication ranges of sensor nodes are disks with the same communication radius [24]; (3) the communication is stable within communication radius [25], and the bandwidth of the links between the mobile sink and all sensor nodes are identical; (4) in a data collection cycle, every sensor node produces the same amount of data, and the mobile sink would spend the same amount of time to collect them; (5) the travel route of the mobile sink is made up of several line segments. Fig. 1 gives an example of the WSN-MS we consider. The notations that are used in the rest of the paper are summarized in Table 1.

Based on these assumptions, we will propose an approach to solve the DLMP whose main idea is to let the mobile sink move and collect data at the same time. The process of data collection is divided into two phases. In the first phase, the mobile sink traverses the whole deployed range of sensor nodes and detects their positions based on the localization method proposed in [16]. In the second phase, the mobile sink performs data collection periodically. To elaborate, the process of data collection includes three steps: the first step is to select some candidate...
<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ω</td>
<td>Plane deployed a WSN-MS.</td>
</tr>
<tr>
<td>s₀</td>
<td>Origin position of mobile sink.</td>
</tr>
<tr>
<td>vₘₐₓ</td>
<td>Maximum speed of mobile sink.</td>
</tr>
<tr>
<td>S</td>
<td>Set of sensor nodes, $S = {s₁, s₂, \ldots, sₙ}$.</td>
</tr>
<tr>
<td>A</td>
<td>Set of anchor points, $A = {a₁, a₂, \ldots, aₖ}$.</td>
</tr>
<tr>
<td>ρ(A)</td>
<td>Travel route of mobile sink, $\rho(A) = (s₀, a'₁, a'₂, \ldots, a'ₖ, s₀)$.</td>
</tr>
<tr>
<td>C(ρ)</td>
<td>Crossover points between travel route and communication ranges of sensor nodes, $C(\rho) = {c₁, c₂, \ldots, cₘ}$.</td>
</tr>
<tr>
<td>ρ(C)</td>
<td>Travel route redefined as the sequence of crossover points $\rho(C) = (s₀, c'₁, c'₂, \ldots, c'ₘ, s₀)$.</td>
</tr>
<tr>
<td>T(ρ)</td>
<td>Line segments of travel route divided by crossover points $T(\rho) = {t₁, t₂, \ldots, tₚ}$ and $tᵢ = l(p(cᵢ), p(cᵢ₋₁))$.</td>
</tr>
<tr>
<td>L(A, ρ)</td>
<td>Delivery latency.</td>
</tr>
<tr>
<td>Lᶜᵖ(A, ρ)</td>
<td>Time that the mobile sink stays on the crossover points.</td>
</tr>
<tr>
<td>Lᶜˢ₉(A, ρ)</td>
<td>Time that the mobile sink travels on the uncovered travel segments.</td>
</tr>
<tr>
<td>Lᶜˢ₅(A, ρ)</td>
<td>Time that the mobile sink travels on the covered travel segments.</td>
</tr>
<tr>
<td>Rᶜᵖₙₓₘ(C)</td>
<td>Matrix denotes the relationship between the set of sensor nodes $S$ and the set of crossover points $C$, $rᶜᵖᵢⱼ = \begin{cases} 1, \ (c(p(sᵢ)) \cap cⱼ \neq ø) \ 0, \ (c(p(sᵢ)) \cap cⱼ = ø) \end{cases}$</td>
</tr>
<tr>
<td>Rᶜˢₙₓₚ(T)</td>
<td>Matrix denotes the relationship between the set of sensor nodes $S$ and the set of travel segments $T$, $rᶜˢᵢⱼ = \begin{cases} 1, \ (d(p(sᵢ)) \cap tⱼ - {cⱼ, cⱼ₋₁} \neq ø) \ 0, \ (d(p(sᵢ)) \cap tⱼ - {cⱼ, cⱼ₋₁} = ø) \end{cases}$</td>
</tr>
<tr>
<td>Rᶜₛₙₓₚ</td>
<td>Matrix denotes the time assigned for sensor nodes whose communication range is traversed by travel segments.</td>
</tr>
<tr>
<td>p(sᵢ)</td>
<td>Position of node $sᵢ$.</td>
</tr>
<tr>
<td>d(p(sᵢ), p(sⱼ))</td>
<td>Distance between $sᵢ$ and $sⱼ$.</td>
</tr>
<tr>
<td>c(sᵢ)</td>
<td>Circle with center $p(sᵢ)$.</td>
</tr>
<tr>
<td>b(sᵢ)</td>
<td>Circular range with center $p(sᵢ)$.</td>
</tr>
<tr>
<td>l(p(sᵢ), p(sⱼ))</td>
<td>Line segment located on position $p(sᵢ)$ and $p(sⱼ)$.</td>
</tr>
<tr>
<td>α(R, i)</td>
<td>Function to sum up the elements of the $i$th row of the matrix $R$.</td>
</tr>
<tr>
<td>β(R, j)</td>
<td>Function to sum up the elements of the $j$th column of the matrix $R$.</td>
</tr>
<tr>
<td>τ(sᵢ)</td>
<td>Time that sensor node $sᵢ$ transmits data to mobile sink in one data collection cycle.</td>
</tr>
<tr>
<td>δ(x)</td>
<td>Function that $δ(x) = \begin{cases} 1, \ (x &gt; 0) \ 0, \ (x = 0) \end{cases}$</td>
</tr>
<tr>
<td>³δ(x)</td>
<td>Function that $³δ(x) = \begin{cases} 1, \ (³δ(x) = 0) \ 0, \ (³δ(x) = 1) \end{cases}$</td>
</tr>
</tbody>
</table>
points other than sensor nodes on the deployed plane as anchor points, the second step is to connect all the anchor points as a travel route, and the third step is to create a visiting schedule and assign visiting time to every sensor node. It is worth mentioning that $S$ is the set of cluster heads in heterogeneous WSNs, and the circular range $b(s_i)$ and transmission time $\tau(s_i)$ of sensors may have different values for different sensor nodes. Fig. 1 shows an example: We select points $a_1, a_2$ and $a_3$ as anchor points, form the travel route following the line segments $s_0a_1, a_1a_2, a_2a_3$ and $a_3s_0$, and program the access time by linear programming for sensor node $s_2$ on line segments $c_2c_3$ and $c_3c_4$, for sensor node $s_4$ on line segments $c_3c_4, c_4a_2, a_2a_3, a_3c_7$ and $c_7c_8$, and for sensor node $s_5$ on line segments $c_7c_8$ and $c_8c_9$.

Formally, we have the following definitions.

**Definition 1 (Anchor Point).** Since the travel route of mobile sink is made up of joint line segments, we define the turning points of line segments in terms of anchor points.

$$A = \{a_1, a_2, \ldots, a_k\}$$

where $A$ is the set of anchor points with $k$ elements, and $a_1, a_2, \ldots, a_k$ are $k$ anchor points. For example, in Fig. 1, $A = \{a_1, a_2, a_3\}$, and the travel route is made up of line segments $s_0a_1, a_1a_2, a_2a_3$, and $a_3s_0$.

**Definition 2 (Travel Route).** The travel route is a sequence of anchor points in $A$, which is denoted as follows.

$$\rho(A) = (s_0, a'_1, a'_2, \ldots, a'_k, s_0)$$

where $\rho(A)$ is a travel route of the mobile sink, $A$ is the selected anchor points, $s_0$ is the origin position of the mobile sink, and $a'_1, a'_2, \ldots, a'_k$ are anchor points from $A$. For simplicity, we denote $\rho(A)$ as $\rho$.

Next, we give a theorem to describe the distribution of anchor points on the shortest travel route.

**Theorem 1 (Anchor Points on the Shortest Travel Route).** If a travel route is the shortest, its anchor points must be at the border of communication range of sensor nodes.

**Proof.** Suppose that the travel route $\rho = (s_0, a'_1, a'_2, \ldots, a_{out}, \ldots, a_{in}, \ldots, a'_k, s_0)$ is the shortest one, where the anchor point $a_{out}$ is out of the communication range.
of sensor nodes, $a_{in}$ is within the communication range of sensor nodes, and others except $s_0$ are right on the border of communication range of sensor nodes. Fig. 2 shows an illustration of this scenario.

Based on the above theorem, we can derive that the travel route $\rho = (s_0, a_1', a_2', \ldots, a_{out}', \ldots, a_{in}, \ldots, a_k', s_0)$ is not the shortest one for $a_{out}$ being out of the communication range of sensor nodes and $a_{in}$ being within the communication range of sensor nodes. We can make the travel route shorter by replacing the travel segments $l(p(a_{out-1}), p(a_{out}))$ and $l(p(a_{out}), p(a_{out+1}))$ by the travel segment $l(p(a_{out-1}), p(a_{out+1}))$. Similarly, if the anchor point $a_{in}$ is within the communication range of sensor nodes, we can also make the travel route shorter by replacing the travel segments $l(p(a_{in-1}), p(a_{in}))$ and $l(p(a_{in}), p(a_{in+1}))$ by the travel segment $l(p(a_{in-1}), p(a_{in+1}))$. The positions of the anchor points $a_{out-1}$, $a_{out+1}$, $a_{in-1}$ and $a_{in+1}$ are critical, therefore, they must be on the border of the communication range of sensor nodes. Otherwise we can find travel segments to replace them to make the travel route shorter.

3.2. Problem Formulation

In this section, we first define the delivery latency and the DLMP, and then give two theorems about the travel route planning complexity and the lower bound of delivery latency.

**Definition 3 (Delivery Latency).** The deliver latency is defined as the sum of the time the mobile sink stays on crossover points, the time it travels on the uncovered
travel segments and the time it travels on the covered travel segments in a data collection cycle, which is expressed by

\[ L(A, \rho) = L^{cp}(A, \rho) + L^{tsu}(A, \rho) + L^{tsc}(A, \rho) \]

where \( L(A, \rho) \) is the delivery latency, \( L^{cp}(A, \rho) \) is the time that the mobile sink stays on the crossover points to collect data of sensor nodes, \( L^{tsu}(A, \rho) \) is the time that the mobile sink travels on the uncovered travel segments, and \( L^{tsc}(A, \rho) \) is the time that the mobile sink travels on the covered travel segments, and at the same time collects the data of sensor nodes within its communication range. The parameter \( A \) is the set of anchor points and \( \rho \) is the travel route.

The mobile sink must stay on crossover points to collect data if there are no travel segments traversing the communication range of sensor nodes. That is to say, if the condition \( \alpha(R_{n \times p}, i) = 0 \) holds, it must assign data transmission time for sensor node \( s_i \) on its crossover points, and the assigned time should be at least \( \tau(s_i) \). Then \( L^{cp} \) can be given by

\[ L^{cp}(A, \rho) = \sum_{i=1}^{n} (\tau(s_i) \cdot \delta(\alpha(R_{n \times p}, i))) \] (1)

For uncovered travel segments, it does not need to assign the data transmission time, thus the mobile sink should move as fast as possible. The time that the mobile sink traverses uncover travel segments is formulated as

\[ L^{tsu}(A, \rho) = \sum_{j=1}^{p} \left( \frac{d(t_j)}{v_{\text{max}}} \cdot \delta(\beta(R_{n \times p}, j))) \right) \] (2)

The mobile sink can visit at least one sensor node on the covered travel segments, though we assume that the mobile sink is equipped only one antenna and can only access one sensor node at one time. To reduce the delivery latency, the mobile sink receives data from sensor nodes, and meanwhile travels along the covered segment. When the mobile sink travels on one covered segment, the mobile sink may access several sensor nodes in a time division approach. The time that the mobile sink travels on the covered travel segments can be denoted by

\[ L^{tsc}(A, \rho) = \sum_{i=1}^{n} \sum_{j=1}^{p} (\tau_{ij}^{tsc}) \] (3)
The time assignment matrix $R_{n \times p}^{ts}$ should satisfy two constraints. The first one is that the total time assigned to sensor node $s_i$ should be more than the time taken by transmitting its data, i.e.,

$$\alpha(R_{n \times p}^{ts}, i) \geq \delta(\alpha(R_{n \times p}^{ts}, i)) \cdot \tau(s_i) \quad (4)$$

The second one is that the total time assigned to travel segment $t_j$ should be more than the time of traversing these segments at the maximum speed, i.e.,

$$\beta(R_{n \times p}^{ts}, j) \geq \delta(\beta(R_{n \times p}^{ts}, j)) \cdot \frac{d(t_j)}{v_{\text{max}}} \quad (5)$$

**Definition 4 (Delivery Latency Minimization Problem).** In a WSN-MS, given mobile sink $s_0$ and a set of sensor nodes $S$ which are deployed on a plane $\Omega$, the DLMP is to minimize the delivery latency $L(A, \rho)$ by selecting proper set of anchor points $A$ and planing better travel route $\rho$, so as to derive crossover points $C$, travel segments $T$, matrix $R_{n \times m}^{cp}$, matrix $R_{n \times p}^{ts}$, and matrix $R_{n \times p}^{ts}$.

Minimize

$$L(A, \rho) = \sum_{i=1}^{n} (\tau(s_i) \cdot \delta(\alpha(R_{n \times p}^{ts}, i))) + \sum_{j=1}^{p} \left(\frac{d(t_j)}{v_{\text{max}}} \cdot \delta(\beta(R_{n \times p}^{ts}, j))\right) + \sum_{i=1}^{n} \sum_{j=1}^{p} r_{ij}^{ts}$$

subject to

$$\prod_{i=1}^{n} \alpha(R_{n \times m}^{cp}, i) > 0 \quad (6)$$

$$\alpha(R_{n \times p}^{ts}, i) \geq \delta(\alpha(R_{n \times p}^{ts}, i)) \cdot \tau(s_i) \quad (7)$$

$$\beta(R_{n \times p}^{ts}, j) \geq \delta(\beta(R_{n \times p}^{ts}, j)) \cdot \frac{d(t_j)}{v_{\text{max}}} \quad (8)$$

$$A \subset \Omega \quad (9)$$

$$1 \leq i \leq n \quad (10)$$

$$1 \leq j \leq p \quad (11)$$

In the above formulation, parameters $A$ and $\rho$ are optimization variables, and parameters $\tau(\cdot), v_{\text{max}}$ and $r$ are constants. Eq. (6) is the direct access constraint
that the mobile sink must access every sensor node in a single hop. Eq. (7) indicates the data transmission constraint that the total time assigned to a sensor node should be more than the time of transmitting its data. Eq. (8) represents the route traverse constraint that the total time assigned to the travel segment should be more than the time of traversing these segments at maximum speed. Constraint (9) shows the definition domain of the set of anchor points $A$.

In the following, we have Theorem 2 that gives the complexity of this problem, and Theorem 3 that gives the lower bound of delivery latency.

**Theorem 2 (Travel Route Planning Complexity).** In DLMP, the decision version of travel route planning problem is an NP-complete problem.

**Proof.** We can reduce any instance of the TSP problem to an instance of the DLMP problem. Since the TSP problem is an NPC problem, the DLMP is also an NPC problem.

The decision version of DLMP and TSP are formally described as follows.

- Decision version of DLMP: Given a graph $G = (V, E)$, does there exist a travel route (visiting sequence) that makes the delivery latency reach at most $L$?

- Decision version of TSP: Given a graph $G' = (V', E')$, does there exist a tour (visiting sequence) that covers all nodes in $V'$ and the tour length is at most $D$?

First, given a travel route $\rho = (s_0, a'_1, a'_2, \ldots, a'_k, s_0)$, we can verify whether the travel time can reach $L(L = L^{cp} + L^{tsu} + L^{tsc})$ by calculating the collection of crossover points $C$, the collection of travel segments $T$, the matrix $R_{n \times m}^{cp}$, the matrix $R_{n \times p}^{ts}$, and the matrix $R_{n \times p}^{tsc}$ in $O(|S| \times |A|)$ time, and solve the linear program of Eqs. (3), (4), and (5) in polynomial time to achieve $L^{tsc}$. Thus, the travel route planning problem is an NP problem.

Second, we can reduce any instance of TSP to an instance of DLMP as follows. Suppose that any instance of TSP is denoted as $(G' = (V', E'), D)$, any instance of DLMP is denoted as $(G = (V, E), A, s_0, r, \gamma(\cdot), v_{\text{max}}, L)$. We can construct the
instance of DLMP from the instance of TSP as follows.

\[ G' = G \]
\[ V = A \cup \{ s_0 \} \]
\[ r = 0 \]
\[ \gamma(.) = 1 \]
\[ v_{\text{max}} = 1 \]
\[ L = D + k \]

If \( \rho = (s_0, a_1', a_2', \ldots, a_k', s_0) \) is a tour of TSP, which covers all nodes in \( V' \) and has a length of at most \( D \). Then, \( \rho = (s_0, a_1', a_2', \ldots, a_k', s_0) = (s_0, a_1'', a_2'', \ldots, a_k'', s_0) \) is a travel route of DLMP. For \( r = 0 \), the communication range of sensor nodes is only a point. Thus, we can derive that

\[
D = \sum_{i=1}^{k-1} \left( \frac{d(a''_i, a''_{i+1})}{v_{\text{max}}} \right) + \frac{d(s_0, a''_1)}{v_{\text{max}}} + \frac{d(a''_k, s_0)}{v_{\text{max}}}
\]

\[
R^{cp}_{k \times k} = \begin{pmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{pmatrix}
\]

\[
R^{ts}_{k \times (k+1)} = \begin{pmatrix}
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0
\end{pmatrix}
\]

\[
L^{cp} = \sum_{i=1}^{k} (\gamma(i)) = k
\]

\[
L^{tsu} = \sum_{i=1}^{k-1} \left( \frac{d(a'_i, a'_{i+1})}{v_{\text{max}}} \right) + \frac{d(s_0, a'_1)}{v_{\text{max}}} + \frac{d(a'_k, s_0)}{v_{\text{max}}}
\]

\[
L^{tsc} = 0
\]

\[ L = k + D \]

Since the TSP problem is an NPC problem, the DLMP problem is also an NPC problem. ■
Theorem 3 (Lower Bound of Delivery Latency). In DLMP, the lower bound of delivery latency is the sum of data transmission time of all sensor nodes.

\[ L \geq \sum_{i=1}^{n} \tau(i) \]  \hspace{1cm} (12)

**Proof.** This theorem implies that the travel time of mobile sink must be equal or greater than the time that the mobile sink gathering data of all sensor nodes. We can derive the result by applying the function \( \alpha(R, i) \) in Table 1 to lines (13)-(14), and applying Eqs. (6) and (7) to lines (14)-(15).

\[
L \geq \sum_{i=1}^{n} (\tau(s_i) \cdot \delta(\alpha(R^{ts}_{nstp, i}))) + \sum_{i=1}^{n} \sum_{j=1}^{p} (r^{ts}_{ij}) \hspace{1cm} (13)
\]

\[
= \sum_{i=1}^{n} (\tau(s_i) \cdot \delta(\alpha(R^{ts}_{nstp, i}))) + \sum_{i=1}^{n} \alpha(R^{ts}_{nstp, i}) \hspace{1cm} (14)
\]

\[
\geq \sum_{i=1}^{n} (\tau(s_i) \cdot \delta(\alpha(R^{ts}_{nstp, i}))) \hspace{1cm} (15)
\]

\[
+ \sum_{i=1}^{n} (\tau(s_i) \cdot \delta(\alpha(R^{ts}_{nstp, i}))) \hspace{1cm} (16)
\]

\[
= \sum_{i=1}^{n} (\tau(s_i) \cdot (\delta(\alpha(R^{ts}_{nstp, i}))) + \delta(\alpha(R^{ts}_{nstp, i})))) \hspace{1cm} (17)
\]

\[
= \sum_{i=1}^{n} \tau(s_i) \hspace{1cm} (18)
\]

Note that in inequality (13), the function \( \alpha(R^{ts}_{nstp, i}) \) is the sum of the elements of the \( i \)th row of the matrix \( R^{ts}_{nstp} \), by the definition in Table 1, i.e., \( \sum_{j=1}^{p} (r^{ts}_{ij}) = \alpha(R^{ts}_{nstp, i}) \), which implies that line (13) holds. Lines (14)-(15) are based on the constraint (7): \( \alpha(R^{ts}_{nstp, i}) \geq \delta(\alpha(R^{ts}_{nstp, i})) \cdot \tau(s_i) \).

4. Heuristic Algorithms

Since the DLMP problem is an NP-complete problem based on Theorem 2, it is impossible to find an effective polynomial time algorithm to solve DLMP. Next, we propose a substitution heuristic algorithm (SHA) to achieve an approximate optimal solution. The main idea of SHA is to use a point substitution process and
a line substitution process to find a shorter travel route, and to use a relaxed linear programming to determine the visiting time of each sensor node and minimize the delivery latency of mobile sink.

4.1. Substitution Process

In SHA, we perform two types of substitutions: point substitution (PS) and line substitution (LS). By Theorem 1, when anchor points are located at the border of communication range of sensor nodes, the length of travel route is shorter. Based on this property, the point substitution is to transfer anchor points from the center to the border of communication range of sensor nodes. The point substitution algorithm is described in Algorithm 1.

Algorithm 1 Point Substitution (PS)

\textbf{Input:}
\hspace{1cm} \rho^{'\prime}: \text{travel route output by classical TSP algorithm}

\textbf{Output:}
\hspace{1cm} \rho: \text{travel route generated by point substitution}

1: \rho_1 \leftarrow s_0;
2: \text{for } i = 2 \text{ to } \|\rho^{'\prime}\| - 2 \text{ do}
3: \rho_i \leftarrow \text{select a point on the border of communication range of sensor node } s_{\rho^{'\prime}};
4: \text{end for}
5: \rho_{\|\rho^{'\prime}\|} \leftarrow s_0;
6: \text{return } \rho;

The line substitution is to make the travel route shorter, whose main idea is to replace two adjacent travel segments with one shorter travel segment based on the principle that the sum of two sides of a triangle are greater than the third one in Euclidean space. The line substitution must satisfy two conditions. One is that the total length of two adjacent travel segments must be larger than the length of the direct connected line segment. The other is that all sensor nodes must be traversed after line substitution. The line substitution algorithm is described in Algorithm 2.

4.2. Visiting Schedule

The visiting schedule of the mobile sink is used to determine the time assignment matrix $R^{tsc}_{n \times p}$ and the travel time on the covered travel segments $L^{tsc}$. From
Eqs. (6)-(11), we can see that only $L^{tsc}$ is a variable. To minimize the delivery latency $L$, we only need to optimize the traversing time on the covered travel segments $L^{tsc}$. We relax Eq. (6) by dropping constraint (6) and keeping constraints (7) and (8), which is formulated as

$$\text{Minimize} \quad L^{tsc} = \sum_{i=1}^{n} \sum_{j=1}^{p} (r_{ij}^{tsc})$$

(19)

$s \cdot t$.

$$\alpha(R_{n \times p}^{tsc}, i) \geq \delta(\alpha(R_{n \times p}^{ls}, i)) \cdot \tau(s_i)$$

(20)

$$\beta(R_{n \times p}^{tsc}, j) \geq \delta(\beta(R_{n \times p}^{ls}, j)) \cdot \frac{d(t_j)}{v_{\text{max}} }$$

(21)

The relaxed equation cannot be solved directly, thus it must be transformed to the canonical form of the linear programming. The canonical form can be given by

$$\text{Minimize} \quad L^{tsc} = C^T X$$

(22)

$s \cdot t$.

$$BX \geq b$$

(23)

$$X \geq 0$$

(24)

In Eq. (22), $X$ is a decision variable vector of the linear programming whose elements correspond to the nonzero elements of the travel segment access matrix $R_{n \times p}^{ls}$, $C^T$ is a coefficient vector whose element values are all 1, and coefficients matrix $B$ and coefficients vector $b$ are transformed by the constraints (20) and (21).

4.3. **Substitution Heuristic Algorithm (SHA)**

The substitution heuristic algorithm SHA mainly includes four steps. The aim of the first step is to select the positions of sensor nodes as the initial positions of anchor points. The aim of the second step is to use a classical TSP heuristic algorithm to generate the initial travel route. The aim of third step is to utilize point substitution and line substitution to minimize the initial travel route. The aim of the fourth step is to determine the visiting schedule by solving a linear
Algorithm 2: Line Substitution (LS)

**Input:**
- $\rho'$: travel route output by point substitution
- $I$: number of iterations
- $S$: set of sensor nodes
- Check(): check whether all sensor nodes can be traversed after line substitution

**Output:**
- $\rho$: travel route generated by line substitution

1: $k \leftarrow 1$;
2: while $(k++) \leq I$ do
3:   $\rho_0 \leftarrow s_0$, $d \leftarrow \text{null}$;
4:   for $i = 2$ to $\|\rho'\| - 1$ do
5:     $d_i \leftarrow d(p(\rho'_{i-1}), p(\rho'_{i+1})) - d(p(\rho_{i-1}), p(\rho_i)) - d(p(\rho_i), p(\rho_{i+1}))$
6:   end for
7:   order $d$ by decreasing;
8:   for $i = 1$ to $\|\rho'\| - 1$ do
9:     if Check($\rho$, $S$) and $d_\rho_i' > 0$ then
10:        $\rho_i' \leftarrow \text{null}$;
11:     else
12:        $\rho_i \leftarrow \rho_i'$;
13:     end if
14:   end for
15:   $\rho = s_0$;
16:   if $\|\rho\| == \|\rho'\|$ then
17:     return $\rho$;
18:   end if
19:   $\rho \leftarrow \rho'$;
20: end while
21: return $\rho$;
programming problem as shown in Eqs. (19)-(21). Fig. 3 demonstrates part of these steps. SHA is described in Algorithm 3.

We now analyze the complexity of SHA, which is mainly determined by the linear programming solver and the classical TSP solver. We assume that $c_1$ denotes the complexity of the linear programming solver, and $c_2$ indicates the complexity of the classical TSP solver. It is clear that the complexity of the point substitution algorithm is $O(n)$ and the complexity of the line substitution algorithm is $O(I \cdot n + I \cdot n^3 + I \cdot n^2)$, where $n$ is the number of sensors in the network and $I$ is the number of iterations of the algorithm. Accordingly, the complexity of the visiting schedule is $O(n + n \cdot p + c_1)$. The complexity of SHA is, therefore, $O(n + I \cdot n + n \cdot p + n^2 + I \cdot n^3 + c_1 + c_2)$. By predigesting, the complexity of SHA is $O(I \cdot n^3 + c_1 + c_2)$.

5. Performance Evaluations

We have conducted extensive simulations to evaluate the performance of the proposed SHA algorithm. In the simulation, we consider the typical homogeneous WSNs although the algorithm is also applicable to heterogeneous WSNs. In this section, we present and analyze the simulation results.
Algorithm 3 Substitution Heuristic Algorithm (SHA)

**Input:**
- \( s_0 \): original position of mobile sink
- \( S \): set of sensor nodes
- \( \tau(.) \): transmission time
- \( v_{\text{max}} \): maximum speed of mobile sink
- \( I \): number of iterations
- \( TSP(.) \): a classical TSP programming solver

**Output:**
- \( \rho \): travel route
- \( L \): delivery latency
- \( R_{tsc}^{n \times m} \): visiting schedule

1. \( A \leftarrow S + \{ s_0 \} \);
2. \( \rho'' \leftarrow TSP(A, s_0) \);
3. \( \rho' \leftarrow \text{PS with parameter } \rho'' \);
4. \( \rho \leftarrow \text{LS with parameters } \rho' \text{ and } I \);
5. \( R_{tsc}^{n \times m} \leftarrow \text{the definition in Table 1 with parameter } \rho \);
6. \( [L_{tsc}, R_{tsc}^{n \times m}] \leftarrow \text{visiting schedule with parameters } R_{tsc}^{n \times m}, \tau(.) \text{ and } v_{\text{max}} \);
7. \( L \leftarrow \text{the formulas in Definition 3 with parameters } L_{tsc} \text{ and } R_{tsc}^{n \times m} \);
8. return \( \rho, L \) and \( R_{tsc}^{n \times m} \);

5.1. *Performance Metrics*

In the simulation, we focus on three performance metrics: delivery latency, route length and energy consumption. The delivery latency is defined in Definition 3 and is formulated as Eq. (3). The route length is defined as the total length of travel segments on a travel route. If travel route \( \rho = (s_0, c'_1, c'_2, \ldots, c'_m, s_0) \), the route length \( \|\rho\| \) can be formulated as follows

\[
\|\rho\| = d(p(s_0), p(c'_1)) + d(p(c'_m), p(s_0)) + \sum_{i=1}^{m-1} d(p(c'_i), p(c'_{i+1}))
\]  \hspace{1cm} (25)

The energy consumption is measured by the energy consumption of sensor node in one data collection cycle. When the mobile sink moves to some positions of traveling route, the mobile sink only collects data of the nearest sensor node. The energy consumption of sensor node on data transmission and the energy consumption of mobile sink on data receiving are measured by Eq. (26) and Eq. (27),
respectively. [26]

\[ E_{Tx}(k, d) = E_{elec} \times k + \epsilon_{amp} \times k \times d^2, \quad (26) \]

\[ E_{Rx}(k) = E_{elec} \times k. \quad (27) \]

5.2. Comparisons

In this subsection, we compare the proposed SHA algorithm with two other naive algorithms: TSP heuristic algorithm (THA) and random heuristic algorithm (RHA). The main difference between SHA and the other two algorithms is how to select the positions of anchor points. In SHA, although the initial positions of anchor points are set to the positions of sensor nodes, the final positions are always located on the border of communication ranges of sensor nodes which are determined by the point substitution and line substitution procedures. In THA, the positions of anchor points are fixed at the positions of sensor nodes, and in RHA, the positions of anchor points are randomly selected within the communication ranges of sensor nodes. Once the anchor points are determined, all the three algorithms use the same TSP programming solver to find the travel routes, and utilize the same procedures to compute the travel segments matrix \( R_{ts}^{n \times p} \), the cross over points matrix \( R_{cp}^{m \times m} \), the initial time assignment matrix \( R_{tsc}^{n \times p} \), and the visiting schedule.

Specifically, for THA, it can be obtained from Algorithm 3 by rewriting Line 2 as \( \rho \leftarrow TSP(A, s_0) \), and dropping Line 3 and Line 4. For RHA, it can be obtained from Algorithm 3 by rewriting Line 1 as \( A \leftarrow R + \{s_0\} \) (where \( R \) is the set of anchor points selected randomly within the communication ranges of sensor nodes), rewriting Line 2 as \( \rho \leftarrow TSP(A, s_0) \), and dropping Line 3 and Line 4.

5.3. Simulation Results

We set default simulation parameters as Table 2. Fig. 4 shows the travel route obtained by THA, RHA, and SHA, respectively. From Fig. 4, we can see that the anchor points of SHA are almost located on the communication range borders of sensor nodes, and its route length is clearly shorter. Fig. 5 shows the route length and the delivery latency on the three algorithms. From Fig. 5, we can see that SHA can obtain shorter route length and smaller delivery latency. Fig. 6 shows the energy consumption on the three algorithms. From Fig. 6, we can find that RHA and THA consume less energy than SHA, but the energy consumption of RHA is not stable. Overall, there are no much differences among the three algorithm. From Fig. 5 and Fig. 6, we can conclude that there is a tradeoff between delivery latency and energy consumption.
Table 2: Default Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega$</td>
<td>$[0, 800] \times [0, 600]$</td>
<td>Plane deployed the WSN-MS</td>
</tr>
<tr>
<td>$S$</td>
<td>${s_1, s_2, \ldots, s_{50}}$</td>
<td>Set of sensor nodes</td>
</tr>
<tr>
<td>$r$</td>
<td>70</td>
<td>Communication radius [25]</td>
</tr>
<tr>
<td>$v_{max}$</td>
<td>1</td>
<td>Maximization speed of mobile sink</td>
</tr>
<tr>
<td>$\tau(.)$</td>
<td>10</td>
<td>Transmission time of sensor node</td>
</tr>
<tr>
<td>$E_{elec}$</td>
<td>$50nJ/bit$</td>
<td>Transmitter electronics and receiver electronics [26]</td>
</tr>
<tr>
<td>$\epsilon_{amp}$</td>
<td>$100pJ/bit/m^2$</td>
<td>Transmit amplifier [26]</td>
</tr>
<tr>
<td>$k$</td>
<td>20kb/s</td>
<td>Data transmission speed of sensor node [26]</td>
</tr>
</tbody>
</table>

Figure 4: Travel routes of THA, RHA and SHA. Route length $\| \rho_{T,HA} \| = 3627.28$, $\| \rho_{R,HA} \| = 2884.81$, $\| \rho_{S,HA} \| = 1885.64$. Delivery latency $L_{T,HA} = 3488.43$, $L_{R,HA} = 2411.01$ and $L_{S,HA} = 1995.47$. For THA, the mean energy consumption of sensor node is 0.1545J, for RHA, the value is 0.1538J and for SHA, the value is 0.1568J.
The impact of the moving speed on the three algorithms is shown in Fig. 7, and in particular, the detailed impact on SHA is shown in Fig. 8. From Fig. 7(a) and Fig. 8(a), we have the following observations. First, the route length will not be affected by the moving speed of the mobile sink. Second, the route length of SHA is always the shortest. Third, the average route length of RHA is not stable. In addition, from Fig. 7(b), we can see that the delivery latency of the three algorithms drops very quickly with the increase of the moving speed. We can also find that the delivery latency of the three algorithms converges to a constant when the speed is greater than 10. In Fig. 8(b), the delivery latency on uncovered travel segments, the delivery latency on covered travel segments and the total delivery latency all drop when the moving speed increases. Although Fig. 7 and Fig. 8 cannot show which algorithm is the best one, they imply that the delivery latency of the three algorithms will get close to the smallest delivery latency when the moving speed becomes higher.

The impact of transmission time on the three algorithms is shown in Fig. 9, and the detailed impact on SHA is shown in Fig. 10. Both Fig. 9(a) and Fig. 10(a) reveal that the route length will not be affected by transmission time. From Fig. 9(b), we have the following observations. First, SHA can achieve shorter deliver latency than other two algorithms when $\tau(\cdot) < 70$. Second, the delivery latency
Figure 6: Energy consumption of THA, RHA and SHA. In THA, the mean value is 0.1545J; in RHA, the mean value is 0.1538J; in SHA, the mean value is 0.1568J.
Figure 7: Impact of moving speed of THA, RHA and SHA.

Figure 8: Impact of moving speed on SHA.
Figure 9: Impact of transmission time of THA, RHA and SHA.

Figure 10: Impact of transmission time on SHA.
will increase as transmission time becomes larger. Third, the delivery latency of the three algorithms will intercross together and then increase linearly when $\tau(\cdot) \geq 70$. By Theorem 3, we can find that the value on the cross line approaches the smallest delivery latency. In Fig. 10(b), $L_{tsu}$, $L_{tsc}$ and $L$ will also increase when the transmission time becomes longer.

The impact of communication radius on different algorithms is shown in Fig. 11. We have the following observations from Fig. 11(a). First, as the communication radius becomes larger, the route length of THA remains unchanged, whereas the route length of SHA decreases. Second, the average route length of RHA decreases with the increase of the communication radius. However, the decrease of SHA is faster than that of RHA. Similarly, the delivery latency renders the same variation when the communication radius becomes larger. Fig. 11(b) shows that the delivery latency of SHA equals the smallest value when $r \geq 700$, and the delivery latency of RHA equals the smallest value when $r \geq 850$.

The detailed impact of communication radius on SHA is shown in Fig. 12. Fig. 12(a) shows that a larger communication radius makes the route length on uncovered travel segments $\|\rho\|_{tsu}$ and the total route length $\|\rho\|$ shorter, and it also makes the route length on covered travel segments $\|\rho\|_{tsc}$ longer firstly and then shorter. In Fig. 12(b), the delivery latency on uncovered travel segments $L_{tsu}$ and the delivery latency $L$ will became shorter when the communication radius
is larger. At last, the total route length will be \( \| \rho \| = 0 \), and the delivery latency \( L = 50 \) which is the smallest value.

6. Conclusions

In this paper, we have studied the delivery latency minimization problem in a wireless sensor network with mobile sink deployed on a plane randomly. To narrate this problem, we first give some definitions, such as anchor point, travel route, and delivery latency. The delivery latency is referred to as the time that the mobile sink traverses the WSN, and at the same time, collects data of sensor nodes. Furthermore, the delivery latency is decomposed into the sum of the time that the mobile sink stays on the crossover points, the time that the mobile sink travels on the uncovered travel segments, and the time that the mobile sink travels on the covered travel segments. Our goal is to minimize the delivery latency, therefore, the delivery latency minimization problem is formulated as an integer programming subject to the direct access constraint, the data transmission constraint and the route traverse constraint. Due to its NPC complexity, we propose a substitution heuristic algorithm (SHA) which utilizes TSP heuristic algorithm to produce the access sequence of anchor points, and uses substitution to reduce the route length of the mobile sink based on the principle that the anchor points should be distributed on the border of the communication range of sensor...
nodes. Moreover, we propose two algorithms TSP heuristic algorithm (THA) and random heuristic algorithm (RHA). From our simulations, we find that there is a tradeoff between delivery latency and energy consumption, the delivery latency of the three algorithms will approach the shortest delivery latency when the moving speed becomes higher, and will decrease when the speed and the communication radius become smaller. By comparing with two other algorithms, THA and RHA, we find that the proposed SHA algorithm outperforms THA and RHA in terms of shortening delivery latency and reducing route length. In addition, the anchor point selection and other transmission issues may affect the transmission latency, which we plan to further study in our future work.

Acknowledgments

This research work was supported in part by the National Natural Science Foundation of China (61170248, 61373179, 61373178, 61402381), Science and Technology Leading Talent Promotion Project of Chongqing (cstc2013kjrc-ljrcj40001), Fundamental Research Funds for the Central Universities (XDJK2013A018, XDJK2013C094, 2362014XK12), the US National Science Foundation under grant numbers ECCS-0801438 and ECCS-1307576 and US Army Research Office under grant number W911NF-09-1-0154.


Author Biographies

Jiqiang Tang received the BEng degree from the School of Computer and Information Science, Southwest Normal University, Chongqing, China, in 2005. He is currently working toward the PhD degree in the College of Computer Science, Chongqing University, Chongqing, China.

Since 2005, he has been with Chongqing University of Technology, Chongqing, China, as a software engineer and a teacher. His research interests include wireless sensor networks and mobile computing.

Hongyu Huang received his B.Engr. degree from the Department of Computer Science, Chongqing Normal University, China, in 2002. He received his M.S. degree from the College of Software, Chongqing University, China, in 2005. Last, he received his Ph.D degree from the Department of Computer Science, Shanghai Jiao Tong University, China, in 2009. Since 2009, he has been with the College of Computer Science at the Chongqing University, where he is currently an associate professor. Dr. Huang has co-authored over 20 research papers, most of them are focusing on the research fields of vehicular ad hoc networks and wireless sensor networks. His current research interests include large-scale distributed systems, cloud computing, and vehicular ad hoc networks.
Songtao Guo received his B.S., M.S. and Ph.D. degrees in Computer Software and Theory from Chongqing University, Chongqing, China, in 1999, 2003 and 2008, respectively. He was a professor from 2011 to 2012 at Chongqing University. At present, he is a full professor at Southwest University, China. He was a senior research associate at the City University of Hong Kong from 2010 to 2011, and a visiting scholar at Stony Brook University, New York, USA, from May 2011 to May 2012. His research interests include wireless sensor networks, wireless ad hoc networks and parallel and distributed computing. He has published more than 30 scientific papers in leading refereed journals and conferences. He has received many research grants as a Principal Investigator from the National Science Foundation of China and Chongqing and the Postdoctoral Science Foundation of China.

Yuanyuan Yang received the BEng and MS degrees in computer science and engineering from Tsinghua University, Beijing, China, and the MSE and PhD degrees in computer science from Johns Hopkins University, Baltimore, Maryland. She is a Professor of Computer Engineering and Computer Science at Stony Brook University, New York, and the Director of Communications and Devices Division at New York State Center of Excellence in Wireless and Information Technology (CEWIT). Dr. Yang's research interests include wireless networks, optical networks, high speed networks, and parallel and distributed computing systems. Dr. Yang has published over 300 papers in major journals and refereed conference proceedings and holds seven U.S. patents in these areas. She is currently the Associate Editor-in-Chief for IEEE Transactions on Computers, and has served as an Associate Editor for IEEE Transactions on Computers and IEEE Transactions on Parallel and Distributed Systems. She is a Fellow of the IEEE.