Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 761274, 10 pages
http://dx.doi.org/10.1155/2015/761274

*Research Article*

# Maximizing the Lifetime of Wireless Sensor Networks Using Multiple Sets of Rendezvous

**Bo Li and Sungkwon Park**

*Department of Electronics and Computer Engineering, Hanyang University, Seoul 133-791, Republic of Korea*

Correspondence should be addressed to Sungkwon Park; sp2996@hanyang.ac.kr

Received 27 May 2015; Revised 20 August 2015; Accepted 7 September 2015

Academic Editor: Bjorn Landfeldt



In wireless sensor networks (WSNs), there is a "crowded center effect" where the energy of nodes located near a data sink drains much faster than other nodes resulting in a short network lifetime. To mitigate the "crowded center effect," rendezvous points (RPs) are used to gather data from other nodes. In order to prolong the lifetime of WSN further, we propose using multiple sets of RPs in turn to average the energy consumption of the RPs. The problem is how to select the multiple sets of RPs and how long to use each set of RPs. An optimal algorithm and a heuristic algorithm are proposed to address this problem. The optimal algorithm is highly complex and only suitable for small scale WSN. The performance of the proposed algorithms is evaluated through simulations. The simulation results indicate that the heuristic algorithm approaches the optimal one and that using multiple RP sets can significantly prolong network lifetime.

## 1. Introduction

A wireless sensor network (WSN) is composed of spatially distributed autonomous sensor nodes that monitor physical or environmental conditions. WSN has a wide field of applications such as intelligent transportation [1], health monitoring [2], military information integration [3], and forest fire detection [4]. In a WSN, sensor nodes generate sensory data which are then delivered to one or more data sinks using multihop forwarding. All sensory data from the entire network ultimately pass through sensor nodes situated at one-hop distance from a sink node. That is, those nodes within one-hop distance to a sink node must transmit all data generated from the entire network. These sensor nodes consume significant energy for data transmission causing rapid energy depletion. Once all available energy of them is depleted, sensory data from the whole network is unable to reach the data sink rendering the network inoperative. This phenomenon is called the "crowded center effect" [5]. In order to mitigate this problem, rendezvous-based approaches are proposed [6–13].

In rendezvous-based approaches, some nodes are selected as rendezvous points (RPs) to gather data from other nodes. Each RP is visited by a mobile device, referred to as a mobile element (ME), which collects the gathered data. A RP transmits this data when the ME passes by and the ME carries the collected data back to the sink. Utilizing this approach, network traffic is distributed to RPs. Designating additional RPs increases the dispersion of network traffic, thereby increasing network lifetime as well. However, nodes cannot be arbitrarily designated as RPs as the ME must visit each RP to collect data and transmit the data to the sink within a delay constraint. If there are too many RPs, the travel time of the ME will exceed the delay constraint.

In a rendezvous-based method, typically a single set of nodes is selected as RPs. Consequently, the selected RPs are constantly subjected to the "crowded center effect" and their energy depletes at a faster rate than regular nodes. In order to further balance energy consumption, we propose implementing *multiple* sets of RPs in turn to gather data. As shown in Figure 1, assume that in a WSN RP sets $\{1, 3\}$ and $\{2, 4\}$ are both feasible. Suppose that we only use RP set $\{1, 3\}$ and allow nodes 2 and 4 to forward their data to nodes 1 and 3, respectively, as shown in Figure 1(a). As a result, nodes 1 and 3 will consume their available energy sooner than nodes 2 and 4. When nodes 1 and 3 expire, the network becomes inoperative despite nodes 2 and 4 having available energy. However, if we set nodes 1 and 3 as RPs as shown in Figure 1(a)

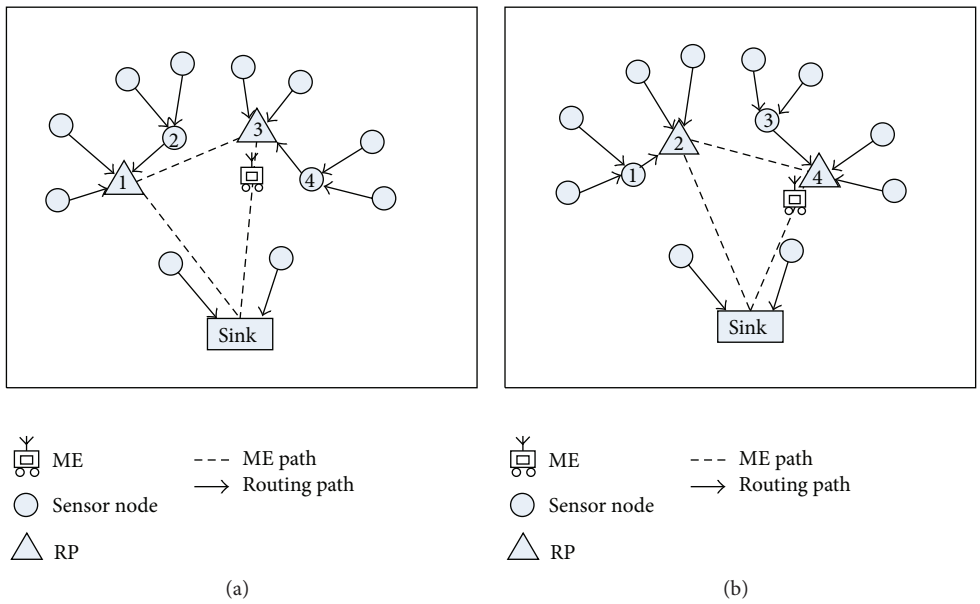


Figure 1: Use multiple sets of RPs in turn to prolong lifetime of WSN further.

only for a portion of the time and then designate nodes 2 and 4 as RPs for the remainder of the time as shown in Figure 1(b), then the average energy consumption of nodes 1 and 3 is reduced and the lifetime of the network is extended. The primary goal is to utilize multiple RPs at different times to average their energy consumption. The problem is how to select the different RP sets and how long to utilize each RP set in order to maximize network lifetime. We first propose an optimal algorithm to address this problem, which identifies all feasible RP sets and selects the optimal time fraction to use each RP set by solving linear programming problems. If there are too many feasible RP sets, the computation complexity of finding the best solution is extremely high. Therefore, we also propose a heuristic algorithm for this situation.

The contributions of this paper are summarized as follows: (1) We formulate the problem of using multiple sets of RPs to maximize the lifetime of WSN; (2) we propose an optimal algorithm for this problem; (3) we propose a heuristic algorithm for large scale WSN with low complexity. The remainder of this paper is organized as follows: Related rendezvous-based energy saving methods for WSN are reviewed in Section 2. Section 3 formally states the problem we studied in this paper. Section 4 presents the proposed algorithms. In Section 5, we evaluate the performance of the proposed algorithms. Finally, conclusions are presented in Section 6.

## 2. Related Work

In WSNs, the mobile element (ME) has been implemented to reduce the energy consumption of sensor nodes. The related approaches can be classified into two types: (1) *one-hop* and (2) *rendezvous-based*. In one-hop approaches, the ME visits each sensor node, and a sensor node only needs to transmit its data to the ME via one hop. Consequently, the energy consumption of sensor nodes is reduced significantly. Initially, researchers proposed using MEs with random mobility such as animals [14], an access point mounted on a bus [15], or a flying mobile agent [16] to collect data from sensor nodes. Although these methods can be implemented with relative ease, random mobility uses time inefficiently for data collection. To address this, controlled mobility approaches were subsequently proposed. Nesamony et al. studied the problem of finding the minimum length path for a ME to visit each sensor node for data collection [17]. The problem is essentially the well-known traveling salesman problem (TSP) [18] which is NP-hard. Therefore, several heuristic algorithms were proposed for the ME path selection [19–25].

Although one-hop approaches can minimize energy consumption of sensor nodes by avoiding multihop transmission, this causes high latency when collecting data from a large number of sensor nodes. In order to reduce data delivery delay, rendezvous-based approaches were proposed. In these approaches, a subset of sensor nodes are selected as RPs to gather data from other nodes via multihop transmission. A ME only visits RPs to collect data, and the data collection time cannot exceed a delay constraint. The problem studied in this paper falls into rendezvous-based approaches.

The rendezvous-based model mentioned above was first implemented in [7, 8]. The authors proved that the problem of selecting a single set of RPs is NP-hard and instead proposed RP-CP and RP-UG algorithms. RP-CP is the optimal algorithm for a special case where the ME only moves along a routing tree which is rooted at the sink. RP-UG, on the other hand, is a heuristic algorithm for the general case. A cluster-based algorithm for rendezvous planning was proposed in [9]. The algorithm iteratively divides sensor nodes into clusters and designates a RP in each cluster. In each iteration, if the length of the shortest tour which covers all RPs is less than the length limit, the algorithm will select additional clusters in the next iteration until the longest feasible ME tour is found. Another algorithm called "rendezvous points

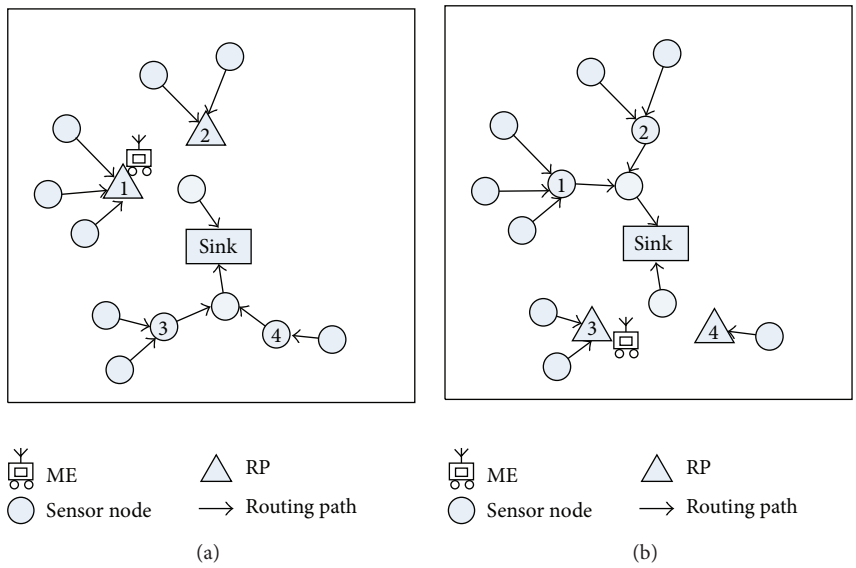


Figure 2: Different data forwarding patterns when different RPs are selected.

selection with load balancing" (RPS-LB) [10] was proposed to jointly optimize path planning for the ME and load balancing for the network. It assumes that a routing tree has been built in the network. From the median of the tree, it traverses the heaviest subtree to construct a tour for the ME. If the tour length is less than the limit, it continues to traverse the second heaviest subtree. This procedure repeats until the tour length reaches the limit which leads to the final solution.

In [11], a greedy method called weighted rendezvous planning (WRP) was proposed to select RPs iteratively. In this method, each node is assigned with a weight equal to the number of packets it must forward per period multiplied by its hop distance from the nearest RP. Initially, only the sink is selected as a RP. In each iteration, the node with the highest weight will be selected as RP and the weight of the remaining nodes is updated accordingly. When the length of the tour which covers the selected RPs exceeds the limit, the procedure ends. Konstantopoulos et al. proposed a clustering method for RP selection based on the watershed transform which is used in image segmentation [12]. The authors enhanced their work by adding an algorithm to periodically reselect a new set of RPs and rebuild the data gathering structure [13]. To the best of our knowledge, this work is the most similar to the methods discussed in this paper.

Other works [26–31] also considered using multiple MEs to collect data from sensor nodes. Authors in [27–29, 31] investigated the problem of designing tours for multiple MEs to gather data with the same deadline, while authors in [27, 29] considered scheduling multiple MEs to harvest sensed data with different deadlines.

## 3. System Model and Problem Statement

In this paper, we consider a WSN with randomly distributed sensor nodes and a single sink node. In order to prolong the lifetime of the WSN, we select *multiple sets* of rendezvous points (RPs) and use them *in turn* to gather data from other sensor nodes. A mobile element (ME) visits currently selected RPs to collect gathered data. Some assumptions about this system are as follows:

(1) The locations of sensor nodes and the sink node are known.

(2) Each sensor node generates one chunk of sensory data with the same size synchronously in every period of $D$.

(3) Each sensor node has an omnidirectional antenna to forward sensory data, and each node has at least one neighbor within its transmission range.

(4) The sensory data generated in each period must be delivered to the sink within $D$ seconds.

(5) The initial energy of all sensor nodes is the same.

(6) A sensor node has enough memory to buffer the data forwarded from its descendants.

For different set of RPs, we need different data forwarding patterns to gather data. For example, Figures 2(a) and 2(b) show the different data forwarding patterns when {node 1, node 2} and {node 3, node 4} are selected as RPs, respectively. In a data forwarding pattern, each RP is a root of a data forwarding tree. If in a data forwarding tree node, data from node A will be forwarded through node B, then we say that node A is a descendant of node B. With different data forwarding patterns, the number of descendants of a node is different. For a given set of RPs, we can let sensor nodes deliver their data to these RPs following different data forwarding patterns. Suppose that there are $M$ possible data forwarding routing patterns in total for all possible RP sets. We denote the number of descendants of node $i$ under $m$th data forwarding pattern by $n_{i,m}$. Consider the data generated by one sensor node during every $D$ seconds as one unit of data. Assume that the energy consumption incurred for

transmitting and receiving one unit of data is $E_{TX}$ and $E_{RX}$, respectively. Under $m$th data forwarding pattern, the energy consumption of node $i$ for data reception in a period is $E_{RX}n_{i,m}$. Node $i$ has to forward the received data and the data generated by itself to its parent node; thus its energy consumption for data transmission in a period is $E_{TX}(n_{i,m}+1)$. The energy consumption of node $i$ in data communication during a period under $m$th data forwarding pattern, denoted by $E_{i,m}$, can be calculated by

$$E_{i,m} = E_{RX}n_{i,m} + E_{TX}\left(n_{i,m}+1\right). \quad (1)$$

We normalize the network lifetime as 1 and denote the fraction of time that $m$th data forwarding pattern is used by $\theta_m$, where $\theta_m \geq 0$ and $\sum_{m=1}^{M}\theta_m = 1$. We only use a part of all possible data forwarding patterns. If $\theta_m = 0$, then $m$th data forwarding pattern is not used. The average energy consumption of node $i$ in data communication during every $D$ seconds is

$$E_i = \sum_{m=1}^{M}\theta_m E_{i,m}. \quad (2)$$

In WSN, most of the energy is consumed by data communication. We consider $E_i$ as the total energy consumption of node $i$ per period. The network lifetime is equal to the lifetime of the node with the highest energy consumption rate. Thus, our goal is to minimize $\max_i E_i$. The main notations in this paper are listed in "Notation" section.

The constraint of the problem is that the sensory data must be delivered to the sink node within $D$ seconds. Suppose that the average velocity of the ME is $v$. The maximum path length of the ME, denoted by $L_{max}$, can be calculated as

$$L_{max} = Dv. \quad (3)$$

Periodically, the ME starts from the sink node to collect data from all RPs and then returns to the sink node. For a set of RPs, we can find the shortest path for the ME to collect data from RPs by a TSP solver [18]. We denote this shortest path length of a RP set $\mathbf{S}$ by PL($\mathbf{S}$), giving us the constraint PL($\mathbf{S}$) $\leq L_{max}$. The distance between any node in $\mathbf{S}$ and the sink must be less than $L_{max}/2$; otherwise PL($\mathbf{S}$) is longer than $L_{max}$. Any node adjacent to the sink should not be included in a RP set $\mathbf{S}$ because it can transmit data to the sink directly and using the ME to collect data from it is an unnecessary energy cost. We set nodes not adjacent to the sink node and with a distance of less than $L_{max}/2$ from the sink as candidates for RPs. For example, as shown in Figure 2, the candidate RPs are nodes 1, 2, 3, and 4. Denoting the set of candidate RPs as $\mathbf{C}$, we can choose any subset $\mathbf{S}$ of $\mathbf{C}$ as RPs at the same time if PL($\mathbf{S}$) $\leq L_{max}$.

Suppose that there are $M$ possible data forwarding patterns for all feasible subsets of $\mathbf{C}$. For each data forwarding pattern, we can calculate $E_{i,m}$ according to (1). Let vector $\mathbf{e}_i = (E_{i,1}, E_{i,2}, \ldots, E_{i,M})^T$ and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_M)^T$. Then, the problem can be formulated as a linear min-max problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \max_i \mathbf{e}_i^T\boldsymbol{\theta}, \\ \text{s.t.} \quad & \sum_{m=1}^{M}\theta_m = 1, \\ & \theta_m \geq 0, \quad 1 \leq m \leq M. \end{aligned} \quad (\text{PI})$$

This problem can be transformed to a set of linear programming problems. We explain this in detail in the next section.

## 4. Proposed Algorithms

In this section, we first propose the optimal algorithm for the problem stated in the previous section. The optimal algorithm has a very high complexity and is only suitable for a small scale WSN. For large scale WSNs, we also propose a heuristic algorithm with a much lower complexity. The heuristic algorithm is more suitable for practical implementations, and we will mainly focus on the heuristic algorithm.

*4.1. The Optimal Algorithm.* The optimal solution for problem (PI) can be found by solving a set of linear programming problems. Consider a problem set (PII) which includes $N$ linear programming problems as follows:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \mathbf{e}_k^T\boldsymbol{\theta}, \\ \text{s.t.} \quad & \mathbf{e}_k^T\boldsymbol{\theta} \geq \mathbf{e}_i^T\boldsymbol{\theta}, \quad i \neq k, \\ & \sum_{m=1}^{M}\theta_m = 1, \\ & \theta_m \geq 0, \quad 1 \leq m \leq M, \\ & (k = 1, 2, \ldots, N). \end{aligned} \quad (\text{PII})$$

Referring to $k$th problem in (PII) as (PII$_k$), we have the following two theorems.

**Theorem 1.** *If* (PII$_k$), $k = 1, 2, \ldots, N$, *has the optimal solution* $\boldsymbol{\theta}_k^*$ *and* $\mathbf{e}_p^T\boldsymbol{\theta}_p^* = \min_{1\leq k\leq N}\mathbf{e}_k^T\boldsymbol{\theta}_k^*$, *then* $\boldsymbol{\theta}_p^*$ *is the optimal solution of* (PI).

*Proof.* See Appendix A. □

**Theorem 2.** *Assume that* $\boldsymbol{\theta}_k^*$ *is the optimal solution of* (PII$_k$); *let* $\mathbf{W} = \mathbf{W}(\boldsymbol{\theta}_k^*) = \{i \mid \mathbf{e}_k^T\boldsymbol{\theta}_k^* = \mathbf{e}_i^T\boldsymbol{\theta}_k^*,\ i \neq k\}$; *then*

(1) *if* $\mathbf{W} = \emptyset$, $\boldsymbol{\theta}_k^*$ *is the optimal solution of* (PI);

(2) *if* $\mathbf{W} \neq \emptyset$ *and, for any* $i \in \mathbf{W}$ $\mathbf{e}_i^T\boldsymbol{\theta}_i^* \geq \mathbf{e}_k^T\boldsymbol{\theta}_k^*$, *then* $\boldsymbol{\theta}_k^*$ *is the optimal solution of* (PI).

*Proof.* See Appendix B. □

Based on these two theorems, we give the optimal algorithm for the problem as follows:

(1) Find all feasible RP sets and all possible data routing patterns for each RP set.

(2) For each routing pattern $m$, calculate $E_{i,m}$, $1 \le i \le N$.

(3) For $k$, $1 \le i \le N$, find the optimal solution $\boldsymbol{\theta}_k^*$ of problem ($\text{PII}_k$).

(4) Find out $\mathbf{W} = \mathbf{W}(\boldsymbol{\theta}_k^*) = \{i \mid \mathbf{e}_k^T\boldsymbol{\theta}_k^* = \mathbf{e}_i^T\boldsymbol{\theta}_k^*,\ i \ne k\}$.

(5) If $\mathbf{W}$ is empty, return $\boldsymbol{\theta}_k^*$; otherwise for each $i \in \mathbf{W}$ find the optimal solution $\boldsymbol{\theta}_i^*$ of problem ($\text{PII}_i$). If $\mathbf{e}_i^T\boldsymbol{\theta}_i^* < \mathbf{e}_k^T\boldsymbol{\theta}_k^*$, set $k = i$ and go to step (4). If for any $i \in \mathbf{W}$ $\mathbf{e}_i^T\boldsymbol{\theta}_i^* \ge \mathbf{e}_k^T\boldsymbol{\theta}_k^*$, return $\boldsymbol{\theta}_k^*$.

Because each candidate RP can be either selected or not in a RP set, given $N$ candidate RPs, there are $2^N - 1$ possible combinations of the them. We check each possible combination to ascertain all feasible RP sets. Also, for each feasible RP set, we must find all possible corresponding routing patterns. If $N$ is large, the computation complexity is extremely high. Therefore, we propose a low complexity heuristic algorithm in the next subsection.

*4.2. The Heuristic Algorithm.* In order to reduce the complexity of the algorithm, we build a routing tree $T$ rooted at the sink node, and each time we select a set of RPs only from nodes at the same level of $T$. Consequently, the search space is significantly reduced. Also, for each set of RPs, we only set one corresponding data forwarding pattern. We construct the routing tree $T$ as follows. We set nodes near the sink node as its children; and for each child of the sink node, we again designate its adjacent nodes as its children. If a node is close to multiple upper level nodes, we set it as a child of the upper level node which has the least children in order to maintain the balance of the tree. This procedure repeats and, ultimately, we build the tree $T$ which covers all sensor nodes. Next, we explain how to select RP sets at each level of $T$. The detailed algorithm is shown in Algorithm 1.

As mentioned before, we do not set nodes near the sink node as RPs. Also, a candidate RP must have a distance of less than $L_{\max}/2$ from the sink node. Denote the level of the candidate RP which has the longest distance from the sink node as $h_{\max}$. We find feasible sets of RPs from level 2 to $h_{\max}$ of tree $T$. The more nodes in a RP set the better. Assume that there are $z$ nodes at level 2. We first try to set all these $z$ nodes as RPs at the same time. If the resulting path length is longer than $L_{\max}$, then we try to select $z-1$ nodes as RPs. If a feasible set of $z-1$ RPs cannot be found, we try to find sets of RPs with $z-2$ nodes and so on. Assume that we can find at least one feasible set of RPs and the number of nodes in the feasible RP set is $z'$. Then, we stop and move to level 3. At level 3, the nodes are farther away from the sink compared to those at level 2. We are unlikely to find a feasible RP set with more nodes than the RP sets found at level 2. In order to reduce complexity, at level 3, we begin to search sets of RPs with $z'$ nodes. Following the same procedure, we can find all feasible RPs at each level.

Assuming that in an iteration we try to find a set of $x$ RPs from $y$ nodes at a level, the detailed method is as follows. As shown in Figure 3, we order the nodes in a clockwise direction. In order to balance the load of RPs, the RPs should be equally spaced so that they will have approximately the same number of descendants. Thus, we always select RPs at regular intervals. Explicitly, as shown from line 4 to line 9 in the algorithm, for each $1 \le i \le \lceil y/x \rceil$, we select a set of nodes $\mathbf{S} = \{i, i + \lceil y/x \rceil, i + 2\lceil y/x \rceil, \ldots, i + (x-1)\lceil y/x \rceil\}$. If a set of nodes $\mathbf{S}$ satisfies the constraint $\text{PL}(\mathbf{S}) \le L_{\max}$, then we set it as a RP set.

For each found RP set at a level $h$, the corresponding data forwarding pattern is set as follows. For nodes not at level $h$, they simply forward their data along the path of the routing tree $T$. For nodes at level $h$, we let them forward their data to the nearest RP rather than to their parents in tree $T$. For example, as shown in Figure 4, assume that nodes 3 and 4 are selected as RPs. Suppose that node 3 is near to nodes 1 and 2 and node 4 is near to node 5. Then, the data forwarding route is $1 \to 2 \to 3$ and $5 \to 4$.

After all RP sets are found, finally we solve linear programming problems to set the time fraction for each RP set as in the proposed optimal algorithm. The computation

**Algorithm to find RPs at different levels of** $T$
**Input:** $T$, locations of sensor nodes, $L_{\max}$
**Output:** $\mathbf{S}_1, \mathbf{S}_2, \ldots$
(1) $x \leftarrow$ number of nodes at level 2, $k \leftarrow 1$
(2) **for** $h = 2$ **to** $h_{\max}$ **do**
(3) $y \leftarrow$ number of nodes at level $h$
(4) order nodes in clockwise direction
(5) **for** $i = 1$ **to** $\lceil y/x \rceil$ **do**
(6) $\mathbf{S} \leftarrow \{i, i + \lceil y/x \rceil, i + 2\lceil y/x \rceil, \ldots, i + (x-1)\lceil y/x \rceil\}$
(7) **if** $\text{PL}(\mathbf{S}) \le L_{\max}$ **then**
(8) $\mathbf{S}_k \leftarrow \mathbf{S}$, $k$++
(9) **end if**
(10) **end for**
(11) **if** no feasible RP set exists **then**
(12) $x \leftarrow x - 1$
(13) **goto** line (4)
(14) **end if**
(15) **end for**

Algorithm 1: The algorithm for finding RPs at different levels of $T$.

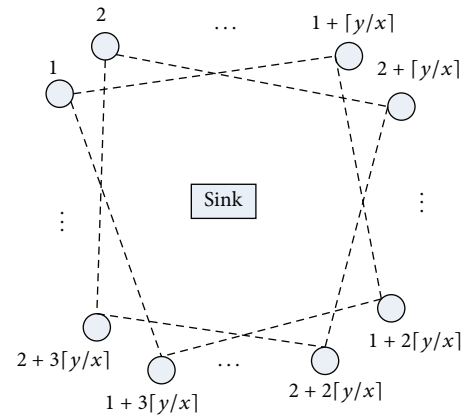


Figure 3: Select one RP in every $\lceil y/x \rceil$ nodes.

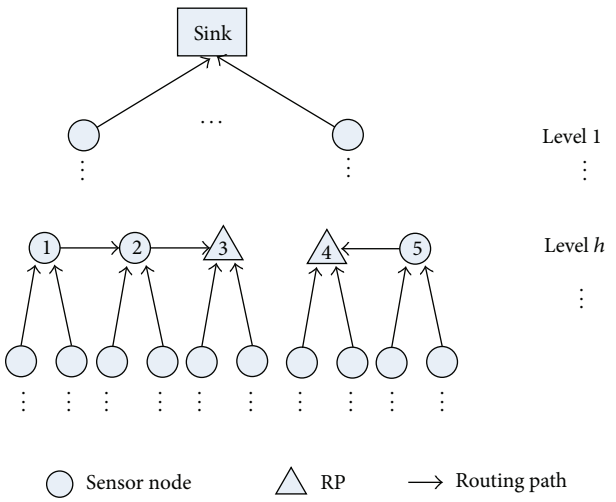


Figure 4: Data forwarding pattern for a RP set at level $h$.

complexity for solving a linear programming problem in problem (PII) is very high if the problem size is large. For example, the algorithm for linear programming in [32] has a time complexity of $O(M(\log M)^{N^2})$, where $M$ is the number of found RP sets (the number of data forwarding patterns equals the number of RP sets because each RP set has only one corresponding data forwarding pattern) and $N$ is the number of candidate RPs. In order to reduce the complexity, we only choose $k_{\max}$ RP sets from $M$ found ones to solve for the final solution where $k_{\max} \le M$. The method to decide the value of $k_{\max}$ will be described in the next subsection, and the way to choose $k_{\max}$ RP sets is as follows. For each selected RP set $m$, $1 \le m \le M$, we calculate $E_{i,m}$, $1 \le i \le N$, according to (1). Let $E_m^{\max} = \max_{1\le k\le N} E_{k,m}$; we sort $M$ RP sets according to $E_m^{\max}$ in ascending order and choose the first $k_{\max}$ RP sets to solve the time fraction $\boldsymbol{\theta}$ for them. Finally, we use this $k_{\max}$ RP sets in turn according to the time fraction allocated to them to prolong the network lifetime.

*4.3. Deciding the Value of $k_{\max}$.* $k_{\max}$ controls the tradeoff between the complexity and performance of the heuristic algorithm. In the heuristic algorithm, solving the linear programming problems to obtain the time fraction $\boldsymbol{\theta}$ is the dominant time-consuming component. As mentioned before, the time complexity of solving the linear programming problems with $M$ RP sets is $O(M(\log M)^{N^2})$, where $N$ is the number of candidate RPs. We choose $k_{\max}$ RP sets. Thus, the complexity of the heuristic algorithm is $O(k_{\max}(\log k_{\max})^{N^2})$. As $k_{\max}$ increases, the complexity of the heuristic algorithm also increases. However, the larger $k_{\max}$ is, the more RP sets will be chosen in the final solution and the more closely the heuristic algorithm approaches the optimal one. The value of $k_{\max}$ should be decided carefully in order to balance the computational complexity and the performance. The algorithm for finding the proper $k_{\max}$ is as follows.

We first set $k_{\max}$ to 1 and then increase it by increments of 1 and estimate the resulting network lifetime. First, we only consider the time fraction allocation for the first two RP sets. Assume $E_{i,1} = \max_{1\le k\le N} E_{k,1}$ and $E_{j,2} = \max_{1\le k\le N} E_{k,2}$; that is, node $i$ and node $j$ have the highest energy consumption in the first and second RP sets, respectively. For node $i$, we should set $\theta_1$ as small as possible. While, for node $j$, we should set $\theta_2$ as small as possible. We have $\theta_1 + \theta_2 = 1$. Thus, we can find the equilibrium between nodes $i$ and $j$ by solving $\theta_1$ and $\theta_2$ in

$$\begin{aligned} \theta_1 E_{i,1} + \theta_2 E_{i,2} &= \theta_1 E_{j,1} + \theta_2 E_{j,2}, \\ \theta_1 + \theta_2 &= 1. \end{aligned} \tag{4}$$

Then we can calculate $E^{(1)} = \theta_1 E_{i,1} + \theta_2 E_{i,2}$. Assume, for the third RP set, $E_{p,3} = \max_{1\le k\le N} E_{k,3}$. We solve equation $\theta'(\theta_1 E_{i,1} + \theta_2 E_{i,2}) + \theta_3 E_{i,3} = \theta'(\theta_1 E_{p,1} + \theta_2 E_{p,2}) + \theta_3 E_{p,3}$, where $\theta' + \theta_3 = 1$, and calculate $E^{(2)} = \theta'(\theta_1 E_{i,1} + \theta_2 E_{i,2}) + \theta_3 E_{i,3}$. This step is repeated iteratively, and, at each iteration $k$, $E^{(k)}$ is an approximation of the optimal average energy consumption with $k+1$ RP sets. As $k$ increases, $E^{(k)}$ approaches the optimal solution. If $E^{(k)} - E^{(k-1)} < \sigma$, where $\sigma$ is a predefined threshold, then we stop and $k_{\max}$ is set to $k + 1$.

## 5. Performance Evaluation

In this section we evaluate the performance of the proposed algorithms through simulations implemented by MATLAB. The proposed optimal and heuristic algorithms are first compared to see how closely the heuristic algorithm can approach the optimal one. The performance of the proposed heuristic algorithm is also compared against two existing methods. One method is proposed by Konstantopoulos et al. in [13] which also dynamically changes RPs. The other method is called WRP [11] in which RPs do not change. In the simulations, we consider a WSN where sensor nodes are randomly distributed in $100 \times 100$ m$^2$ field and the location of the sink node is also randomly chosen. Each simulation was run under 100 different topologies, and we show the average results. In each simulation, the speed of the ME was set to 1 m/s according to the practical experience in packBot system [33], and parameters of sensor nodes were set according to the data sheet of the CC1000 radio on MICA2 motes [34]. Each sensor node generates 2 bytes of sensory data per second, and the transmission rate of it is 40 kb/s. The transmission range of a sensor node was set to 20 m. We assume that the initial energy of all nodes is 100 J, and the energy consumption of the transmitter and receiver circuit of a node is 40 mW and 25 mW, respectively. The parameters are summarized in Table 1. In the simulations, a local-search-based heuristic TSP solver [35] was used.

*5.1. Comparison of the Proposed Optimal and Heuristic Algorithms.* First, the performance of the proposed optimal and heuristic algorithms was compared. Since the computation complexity of the optimal algorithm is very high, we evaluated the performance in a small scale WSN. In this simulation, packet delivery delay $D$ is 100 s and the number of sensor nodes is 50. We set the threshold $\sigma$ to different values

Table 1: Simulation parameters.

| Parameter | Value |
|---|---|
| ME speed | 1 m/s |
| Sensor node transmission range | 20 m |
| Sensor node data sampling rate | 2 B/s |
| Sensor node transmission rate | 40 kb/s |
| Energy consumption of transmitter | 40 mW |
| Energy consumption of receiver | 25 mW |
| Initial energy of sensor nodes | 100 |

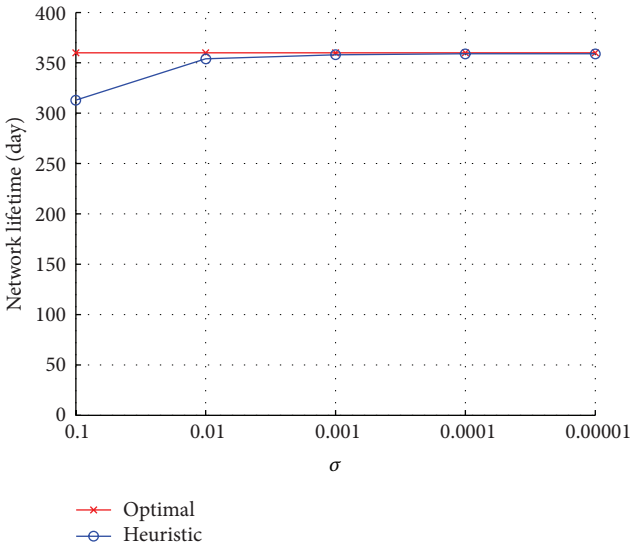


Figure 5: The optimal algorithm versus the heuristic algorithm with different $\sigma$.

to measure its impact on the performance and complexity of the heuristic algorithm and find a proper value for $\sigma$. The simulation results are shown in Figures 5 and 6.

In Figure 5, we can see that as the threshold $\sigma$ decreases the performance of the heuristic algorithm approaches the optimal one. When $\sigma$ decreases from $10^{-3}$ to $10^{-4}$, there is almost no performance improvement. However, as we can see in Figure 6, the corresponding value of $k_{\max}$ increases sharply as $\sigma$ decreases from $10^{-3}$ to $10^{-4}$; and when $\sigma$ reaches $10^{-4}$, $k_{\max}$ no longer increases because $k_{\max}$ already equals the number of all found RP sets. Consequently, setting $\sigma$ to $10^{-4}$ will result in an excess of computation and $10^{-3}$ is a proper value for $\sigma$. We suggest setting the threshold $\sigma$ to $10^{-3}$ in real implementations.

*5.2. Comparison of the Proposed Heuristic Algorithm and Existing Algorithms.* Next, we compare the proposed heuristic algorithm with WRP [11] and the algorithm proposed by Konstantopoulos et al. [13] in a larger scale WSN. The algorithm proposed by Konstantopoulos et al. reselects RPs periodically. We set this period to 5000 seconds. The value of $\sigma$ and the maximum packet delivery delay were set to $10^{-3}$ and 100 s, respectively. First we evaluate the performance with different numbers of sensor nodes. As shown in Figure 7, the

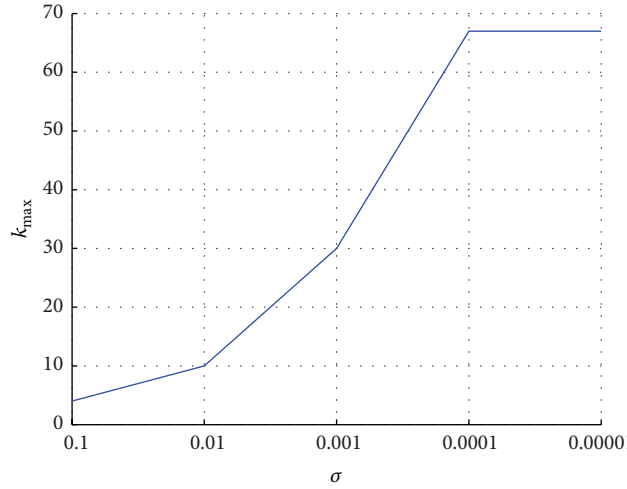


Figure 6: $k_{\max}$ with different $\sigma$.

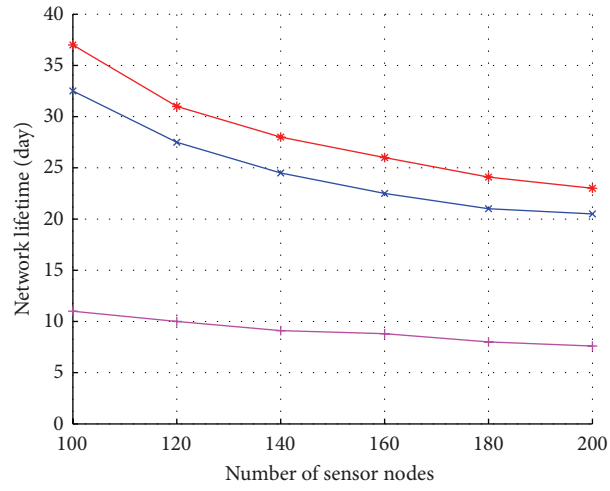


Figure 7: Impact of the number of sensor nodes.

simulation result shows that the performance of the proposed heuristic algorithm is better than that of the two existing ones. On average, the network lifetime achieved by our algorithm is about 3 times of that of WRP and about 10% longer than that of the algorithm of Konstantopoulos et al. [13].

We fixed the number of sensor nodes to 200 and ran the simulation again with different values for the packet delivery delay limit $D$ to see its impact. The simulation result is shown in Figure 8. When $D$ is too low (less than 40 s in the simulation) for the ME to visit a node at level 2, only nodes at level 1 can be set as RPs, which serves no purpose since they can transmit data to the sink directly. As $D$ increases, nodes farther away from the sink can be set as RPs and more RPs can share the traffic at the same time. As a result, the energy consumption of RPs decreases and the network lifetime is extended. With a longer $D$, the benefit of using multiple sets of RPs is more apparent. This is because we can find more

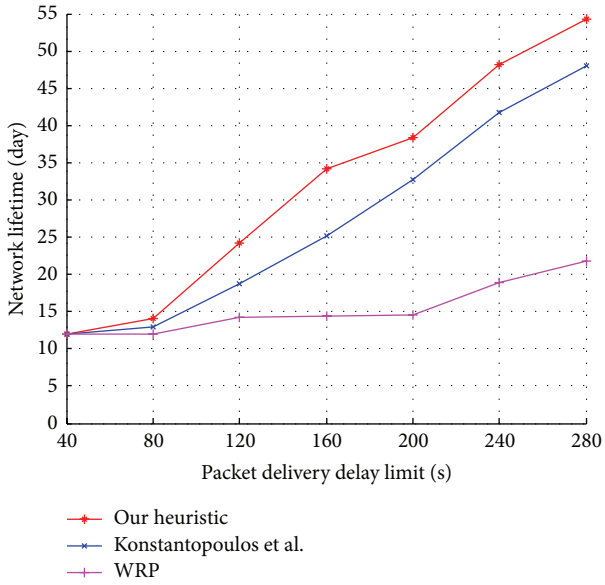


FIGURE 8: Impact of the packet delivery delay limit.

feasible RP sets to average the energy consumption of RPs. As we can see, when $D$ is from 120 s to 200 s, the network lifetime of the proposed heuristic algorithm is about 2 times of that of WRP and the performance gain is about threefold when $D$ is from 200 s to 280 s. The performance of our algorithm is also about 10% higher than that of the algorithm proposed by Konstantopoulos et al. [13].

## 6. Conclusion

In this paper, we studied the problem of using multiple sets of RPs in turn to maximize the lifetime of WSN. We proposed an optimal algorithm and a heuristic algorithm for this problem. The computation complexity of the optimal algorithm is very high and is only suitable for small scale WSN. The heuristic algorithm can approach the optimal performance with much lower complexity in a large scale WSN, and we can adjust the threshold $\sigma$ to balance its complexity and performance. The threshold $\sigma$, the number of sensor nodes, and the packet delivery delay $D$ have a strong impact on the performance of the proposed algorithms. Simulation results show that $10^{-3}$ is a proper value for the threshold $\sigma$. Additionally, the heuristic algorithm approaches the optimal one and can achieve a longer network lifetime than WRP as well as the algorithm proposed by Konstantopoulos et al. [13]. When the number of sensor nodes is 200 and $D$ is from 200 s to 280 s, the network lifetime of the heuristic algorithm is about 3 times that of WRP and about 10% longer than that of Konstantopoulos et al.

## Appendices

## A. Proof of Theorem 1

For any feasible solution $\boldsymbol{\theta}$ of (PI), assume $\mathbf{e}_k^T\boldsymbol{\theta} = \max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta}$; then $\mathbf{e}_k^T\boldsymbol{\theta} \ge \mathbf{e}_i^T\boldsymbol{\theta}$, $i \ne k$; that is, $\boldsymbol{\theta}$ is a feasible solution of ($\text{PII}_k$). Because $\boldsymbol{\theta}_k^*$ is the optimal solution of ($\text{PII}_k$), we have $\mathbf{e}_k^T\boldsymbol{\theta} \ge \mathbf{e}_k^T\boldsymbol{\theta}_k^*$. Because $\mathbf{e}_p^T\boldsymbol{\theta}_p^* = \min_{1\le k\le N} \mathbf{e}_k^T\boldsymbol{\theta}_k^*$ and $\boldsymbol{\theta}_p^*$ is a feasible solution of ($\text{PII}_p$), we have $\max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta} = \mathbf{e}_k^T\boldsymbol{\theta} \ge \mathbf{e}_k^T\boldsymbol{\theta}_k^* \ge \mathbf{e}_p^T\boldsymbol{\theta}_p^* = \max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta}_p^*$. Thus, $\boldsymbol{\theta}_p^*$ is the optimal solution of (PI).

## B. Proof of Theorem 2

(1) If $\mathbf{W} = \mathbf{W}(\boldsymbol{\theta}_k^*) = \emptyset$, at $\boldsymbol{\theta}_k^*$ we have $\mathbf{e}_k^T\boldsymbol{\theta}_k^* \ge \mathbf{e}_i^T\boldsymbol{\theta}_k^*$, $i \ne k$. Because $\mathbf{e}_i^T\boldsymbol{\theta}$ is a continuous function, there exists a neighborhood of $\boldsymbol{\theta}_k^*\mathbf{U}(\boldsymbol{\theta}_k^*)$; that is, for $\boldsymbol{\theta} \in \mathbf{U}(\boldsymbol{\theta}_k^*)$, we have $\mathbf{e}_k^T\boldsymbol{\theta} > \mathbf{e}_i^T\boldsymbol{\theta}$, $i \ne k$, which means $\boldsymbol{\theta}$ is a feasible solution of ($\text{PII}_k$). Thus, for any $\boldsymbol{\theta} \in \mathbf{U}(\boldsymbol{\theta}_k^*)$ we have $\max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta} = \mathbf{e}_k^T\boldsymbol{\theta} \ge \mathbf{e}_k^T\boldsymbol{\theta}_k^* = \max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta}_k^*$; that is, $\boldsymbol{\theta}_k^*$ is a local optimal solution. Because the feasible region of (PI) is a convex set and the objective function of (PI) is a convex function, according to the theory of convex optimization, any local optimal solution of (PI) is the global optimal solution. Thus, $\boldsymbol{\theta}_k^*$ is the optimal solution of (PI).

(2) From part (1) we know $\max_{i\notin\mathbf{W}} \mathbf{e}_i^T\boldsymbol{\theta} \ge \max_{i\notin\mathbf{W}} \mathbf{e}_i^T\boldsymbol{\theta}_k^*$; and because, for any $i \in \mathbf{W}$, $\mathbf{e}_i^T\boldsymbol{\theta}_i^* \ge \mathbf{e}_k^T\boldsymbol{\theta}_k^*$, according to Theorem 1, we have $\max_{i\in\mathbf{W}} \mathbf{e}_i^T\boldsymbol{\theta} \ge \max_{i\in\mathbf{W}} \mathbf{e}_i^T\boldsymbol{\theta}_k^*$. Thus, for any $1 \le i \le N$, we have $\max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta} \ge \max_{1\le i\le N} \mathbf{e}_i^T\boldsymbol{\theta}_k^*$. This completes the proof.

## Notation

| | |
|---|---|
| $C$: | The set of candidate RPs |
| $N$: | The number of candidate RPs |
| $M$: | The number of all possible data forwarding patterns |
| $L_{\max}$: | The maximum tour length of the ME |
| $v$: | The average velocity of the ME |
| $\mathbf{S}_m$: | $m$th RP set |
| $\theta_m$: | The fraction of time to use $m$th data forwarding pattern |
| PL($\mathbf{S}$): | The shortest length of the path made by ME which starts from and ends at the sink node and covers all nodes in set $\mathbf{S}$ |
| $E_{\text{TX}}$: | The energy consumption for transmitting one unit of data |
| $E_{\text{RX}}$: | The energy consumption for receiving one unit of data |
| $E_i$: | The average energy consumption of node $i$ during one period |
| $n_{i,m}$: | The number of descendants of node $i$ when $m$th RP set is used. |

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC

(Convergence Information Technology Research Center) (IITP-2015-H8601-15-1005), supervised by the IITP (Institute for Information and Communications Technology Promotion).

## References

[1] W. Chen, L. Chen, Z. Chen, and S. Tu, "WITS: a wireless sensor network for intelligent transportation system," in *Proceedings of the 1st International Multi—Symposiums on Computer and Computational Sciences (IMSCCS '06)*, vol. 2, pp. 635–641, Hangzhou, China, April 2006.

[2] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems*, pp. 13–24, Baltimore, Md, USA, November 2004.

[3] S. Diamond and M. Ceruti, "Application of wireless sensor network to military information integration," in *Proceedings of the 5th IEEE International Conference on Industrial Informatics*, vol. 1, pp. 317–322, IEEE, Vienna, Austria, June 2007.

[4] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo, "Forest fire detection system based on wireless sensor network," in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*, pp. 520–523, Xi'an, China, May 2009.

[5] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, and I. Stoica, "Balancing traffic load in wireless networks with curveball routing," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 170–179, September 2007.

[6] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 958–973, 2006.

[7] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in wireless sensor networks with mobile elements," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1430–1443, 2008.

[8] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 231–240, ACM, Hong Kong, May 2008.

[9] K. Almi'ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10)*, pp. 582–589, Denver, Colo, USA, October 2010.

[10] L. Mai, L. Shangguan, C. Lang et al., "Load balanced rendezvous data collection in wireless sensor networks," in *Proceedings of the 8th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '11)*, pp. 282–291, IEEE, Valencia, Spain, October 2011.

[11] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2407–2419, 2014.

[12] C. Konstantopoulos, B. Mamalis, G. Pantziou, and V. Thanasias, "Watershed-based clustering for energy efficient data gathering in wireless sensor networks with mobile collector," in *Proceedings of the 18th International Conference on Parallel Computing (Euro-Par '12)*, pp. 754–766, Rhodes Island, Greece, August 2012.

[13] C. Konstantopoulos, G. Pantziou, N. Vathis, V. Nakos, and D. Gavalas, "Efficient mobile sink-based data gathering in wireless sensor networks with guaranteed delay," in *Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '14)*, pp. 47–54, ACM, Québec, Canada, September 2014.

[14] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[15] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Information Processing in Sensor Networks: Second International Workshop, IPSN 2003, Palo Alto, CA, USA, April 22-23, 2003 Proceedings*, Lecture Notes in Computer Science, pp. 129–145, Springer, Berlin, Germany, 2003.

[16] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proceedings of the IEEE Military Communications Conference (MILCOM '03)*, pp. 688–693, Boston, Mass, USA, October 2003.

[17] S. Nesamony, M. K. Vairamuthu, and M. E. Orlowska, "On optimal route of a calibrating mobile sink in a wireless sensor network," in *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS '07)*, pp. 61–64, Braunschweig, Germany, June 2007.

[18] S. Arora, "Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems," *Journal of the ACM*, vol. 45, no. 5, pp. 753–782, 1998.

[19] R. Sugihara and R. K. Gupta, "Improving the data delivery latency in sensor networks with controlled mobility," in *Distributed Computing in Sensor Systems*, vol. 5067 of *Lecture Notes in Computer Science*, pp. 386–399, Springer, Berlin, Germany, 2008.

[20] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 127–139, 2010.

[21] Y. Gu, D. Bozdag, E. Ekici, F. Özgüner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *Proceedings of the Annual 2nd IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 386–395, September 2005.

[22] K. Almi'ani, M. Aalsalem, and R. Al-Hashemi, "Data gathering for periodic sensor applications," in *Proceedings of the 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '11)*, pp. 215–220, IEEE, Gwangju, South Korea, October 2011.

[23] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS '08)*, pp. 1–9, Miami, Fla, USA, April 2008.

[24] K. Almi'ani, M. A. Abuhelaleh, and A. Viglas, "Length-constrained and connected tours for sensor networks," in *Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications, and Technologies (PDCAT '12)*, pp. 105–110, IEEE, Beijing, China, December 2012.

[25] M. A. Abuhelaleh, K. Almi'Ani, and A. Viglas, "Connected tours for sensor networks using clustering techniques," in *Proceedings of the 22nd Wireless and Optical Communications Conference (WOCC '13)*, pp. 432–437, May 2013.

[26] B. A. Alqaralleh and K. Almi'ani, "Mobile elements scheduling for periodic sensor applications," *International Journal of Wireless & Mobile Networks*, vol. 6, no. 1, pp. 15–30, 2014.

[27] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, pp. 395–410, 2007.

[28] L. He, J. Pan, and J. Xu, "Data collection latency in wireless sensor networks with multiple mobile elements," *Ad-Hoc & Sensor Wireless Networks*, vol. 18, no. 1-2, pp. 109–129, 2013.

[29] Y. Gu, D. Bozdağ, R. W. Brewer, and E. Ekici, "Data harvesting with mobile elements in wireless sensor networks," *Computer Networks*, vol. 50, no. 17, pp. 3449–3465, 2006.

[30] K. Almi'ani, A. Viglas, and M. Aalsalem, "Mobile element path planning for gathering transit-time constrained data," in *Proceedings of the 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '11)*, pp. 221–226, IEEE, Gwangju, Republic of Korea, October 2011.

[31] K. Almi'Ani and A. Viglas, "Designing connected tours that almost cover a network," in *Proceedings of the 14th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '13)*, pp. 281–286, Taipei, Taiwan, December 2013.

[32] N. Megiddo, "Linear programming in linear time when the dimension is fixed," *Journal of the ACM*, vol. 31, no. 1, pp. 114–127, 1983.

[33] C. Lundberg, H. I. Christensen, and R. Reinhold, "Long-term study of a portable field robot in urban terrain," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 625–650, 2007.

[34] J. L. Hill and D. E. Culler, "Mica: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, 2002.

[35] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.