Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 801831, 9 pages
http://dx.doi.org/10.1155/2015/801831

Research Article

Designing Constrained Trajectory Based on Maximizing Energy Reduction in Large-Scale Wireless Sensor Networks

Jia Xu,[1,2,3] Chuan Ping Wang,[1] Hua Dai,[1,2] Da Qiang Zhang,[4] and Jing Jie Yu[3]

[1] *School of Computer Science & Technology, Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Jiangsu, Nanjing 210003, China*
[2] *Lianyungang Institute, Nanjing University of Science and Technology, Jiangsu, Lianyungang 222006, China*
[3] *Department of Information Technology, Nanjing General Hospital of Nanjing Military Command, Jiangsu, Nanjing 210002, China*
[4] *School of Software Engineering, Tongji University, Shanghai 201804, China*

Correspondence should be addressed to Jia Xu; xujia@njupt.edu.cn

Received 14 April 2015; Revised 15 August 2015; Accepted 27 August 2015

Academic Editor: Kameswara Namuduri

The *Mobile Sink* based data collection in wireless sensor network can reduce energy consumption efficiently and has been a new data collection paradigm. In this paper, we focus on exploring polynomial algorithm to compute the constrained trajectory of the *Mobile Sink* for data collection. We first present a universal system model for designing constrained trajectory in large-scale wireless sensor networks and formulate the problem as the *Maximizing Energy Reduction for Constrained Trajectory* (MERC) problem. We show that the MERC problem is NP-hard and design an approximation algorithm (CTMER), which follows the greedy approach to design the movement trajectory of the *Mobile Sink* by maximizing the *effective average energy reduction*. Through both rigid theoretical analysis and extensive simulations, we demonstrate that our algorithm achieves high computation efficiency and is superior to other *Mobile Sink* based data collection methods in aspects of energy consumption and network lifetime.

1. Introduction

With the development of digital electronics, wireless communications, and network technology, the applications of wireless sensor networks [1, 2] (referred to as WSNs) are becoming more widespread, including monitoring, event detection, and target tracking. In these applications, the sensor nodes mainly perform three aspects of the jobs: (1) collecting the required data from the environment sensing; (2) the data store or process; and (3) sending or forwarding data from other nodes to the data collection node ultimately, for example, sink node or base station.

In the traditional static wireless sensor network topology, the position of sink node is fixed; thus, the sink node and its surrounding sensor nodes need to transmit a larger amount of data and are likely to die prematurely due to energy exhaustion, which will be the energy bottleneck of large-scale WSNs [3]. In the static network topology, dealing with the energy problem is intractable in sensor network. In recent years, the data collection mode based on the *Mobile Sinks* has been proposed, in which one or more *Mobile Sinks* moved along a specific path in order to reduce the amount of forwarding. This is an efficient way to reduce the energy consumption of the sensor node itself and to make the energy consumption more evenly in large-scale network. In large-scale wireless sensor network, most nodes need to forward data through multihop; thus, the problem of how to reduce the number of forwarding hops in order to reduce the energy consumption has become a key issue. In the extreme case, the *Mobile Sink* can visit each node in WSNs (called flat collection), and the distance between each node and the *Mobile Sink* is only one hop. Obviously, the sensor nodes have the minimum energy consumption in this case; however, the delay of data collection will be the maximum; thus, the flat collection is not suitable for large-scale sensor networks. There is a possible solution which tolerates a certain delay to reduce the energy consumption. Many efforts [4–13] have been made to design the trajectory of the *Mobile Sink*. However, few of them focus on maximizing energy reduction.

In this paper, we design a data collection method based on the *Mobile Sink* to maximize the energy reduction in large-scale WSNs, in which the set of sensor nodes is divided into the sets of *collection trees*. The roots of the *collection tree* are called *collection node*, and the remaining sensor nodes forward the data to the corresponding *collection node*. The *Mobile Sink* visits all *collection nodes* to obtain data from the whole network eventually.

The key contributions of our work are summarized in the following:

(i) We present the system model for data collection in large-scale wireless sensor networks and formulate the movement trajectory design problem as *Maximizing Energy Reduction for Constrained Trajectory* (MERC) problem.

(ii) We show the *MERC* problem is NP-hard and develop an approximation algorithm, *Constrained Trajectory based on Maximizing Energy Reduction* (CTMER), which follows a greedy approach with polynomial time.

(iii) We conduct thorough simulations to investigate the performance of CTMER. The simulation results show that CTMER is superior to other hierarchical data collection methods in aspects of energy consumption and network lifetime.

2. Related Work

The data collection methods based on the mobile infrastructure have been widely studied in recent years. Shah proposed a protocol using Data Mule [14] to collect sensor data in the sparse WSNs. In the scenario of Data Mule, any moving object with the communication function can be used as the mobile infrastructure, such as human, animal, or vehicle with the communication device. As the mobile infrastructures need to collect the data from all the sensors, they should have more energy and larger storage space.

Some universities and research institutions designed the mobile machines or collection vehicles for the data collection in WSNs. University of Southern California designed the mobile node Robomote [15], and Yale University designed the mobile node XYZ [16]. In addition, UCLA designed Packbot [17], and Richard et al. designed NIMS [18]. Zhang et al. designed the MDC—DataTruck [19]. These nodes can be used as the *Mobile Sinks*.

Paper [4] first proposed the concept of predicting the mobility of the *Mobile Sink* to improve performance of the sensor network. The sink nodes are placed on the bus cycle operation, and the sensor nodes are randomly distributed in the sides of the bus operation route. Due to the limit of the movement trajectory of the sink node and the effective communication distance, not all of the sensor nodes can communicate with the sink node directly in the practical applications.

Paper [5] described a road monitoring application and analyzed the trade-off between the data collection and energy in the sparse sensor networks. But it also adopts the single-hop communication as the data transfer mode of nodes.

Paper [6] proposed a data collection method, MASP, which generates the shortest path with the maximum amount of data. MASP focuses on optimization of the relationship between the members of the nodes in the network and the *Mobile Sink* and adopts the 0-1 linear programming to formalize the MASP problem. The authors proposed a genetic algorithm based on two-dimensional chromosome encoding and designed the corresponding data communication protocol. However, it ignores the movement trajectory of the *Mobile Sink*. Moreover, the time complexity can be very high.

Paper [7] proposed a learning based time-domain routing algorithm (HLETDR). The algorithm emphasized that the *Mobile Sink* accesses the fixed sensor nodes (moles) periodically, and the probability of the *Mobile Sink* at the current time domain is predicted in the moles. However, HLETDR did not give the solution to deal with latency problem of the data collection.

RD-VT [8] is an approach to design base station (BS) tour through constructing MST (Minimum Spanning Tree) and SMT (Steiner Minimum Tree). The objective of RD-VT is to find a tour of the BS that visits a set of nodes referred to as *rendezvous points* (RPs) with constrained length of BS tour while minimizing the total Euclidean distance of edges in the routing trees.

The Line-Based Data Dissemination (LBDD) [9] supposes that there are multiple sinks moving randomly in the sensor field and defines a vertical line or strip, which divides the sensor field into two equal parts. The nodes within the boundary of this wide line are called *inline-nodes*. This line acts as a rendezvous area for data storage and lookup. When a sensor detects a new event, it transmits a data report towards the virtual line. This data is stored on the first *inline-node* encountered. To collect the generated data reports, the sink sends its query towards the rendezvous area. The query is then propagated along the virtual line until arriving to the *inline-node* that owns the requested data. However, LBDD is unsuitable for periodic data collection scenarios due to the communication cost.

In Quad tree-based Data Dissemination (QDD) [10] protocol, a common hierarchy of data forwarding nodes is created by the data forwarding nodes using quad tree-based partitioning of the physical network into successive quadrants. In this approach, when a source node detects a new event, it calculates a set of *rendezvous points* by successively partitioning the sensor field into four equally logical quadrants. And the data reports are sent to the nodes which are closer to the centroid of each successive partition. The *Mobile Sink* follows the same strategy for the query packet transmission. The main drawback of this approach is that the *rendezvous points* in high lever will suffer high overhead, and the related hot spot problem may decrease the network lifetime and reliability. Moreover, there is no length limit of the *Mobile Sink* trajectory or deadline of data collection. This assumption may be not reasonable in practice.

In paper [13], the Combine-Skip-Substitute (CSS) scheme is proposed to minimize the tour length of mobile elements. In CSS, all sensor nodes are covered by the tour within a distance of d. However, CSS does not take the length constraint of tour into account and thus cannot be applied to delay sensitivity scenarios.

The Mobility assisted Energy efficient Georouting in energy harvesting Actuator and sensor Networks (MEGAN) [20] is a geographical routing, which reduces energy consumption through moving the sensor nodes toward the positions in the forward direction. However, not all movements of sensor nodes can be controlled in practice.

Paper [21] studies the problem of controlling sink mobility in deadline-based and event-driven applications to achieve maximum network lifetime and proposes an algorithm based on a decision tree and dynamic programming to approximately determine an optimal deadline-based trajectory (ODT). The algorithm assumes that each sensor node can adjust its transmission range; however, this assumption may be invalid in many applications.

3. System Model and Problem Formulation

3.1. System Model. We consider a set of static sensor nodes $V = \{v_1, v_2, v_3, \dots, v_n\}$ randomly distributed in $m \times m$ area densely. Each sensor node v_i, $i = \{1, \dots, n\}$, can obtain its position information. There is a *Mobile Sink* in the network. The *Mobile Sink* collects data along a specific trajectory M and moves with a constant speed C, which is significantly slower than the data transmission speed. The more detailed descriptions about the value of C can be found in *Robomote* [15], *NIMs* [18], and *Packbot* [17]. In every time period T, each sensor node will produce q bit data, which must be forwarded to the *Mobile Sink*, and then the *Mobile Sink* returns to the original position within the T. To reduce the energy consumption and increase the network scalability, a set of *collection nodes*, $V_S = \{v_S^1, v_S^2, \dots, v_S^{|V_S|}\}$, are selected from the sensor node set V in the network. The remaining set of sensor nodes is called *normal node* set: $V_R = \{v_R^1, v_R^2, \dots, v_R^{|V_R|}\}$, $V = V_S \cup V_R$. Any *normal node* $v_R^i \in V_R$ needs to send data to the corresponding *collection node* $S(v_R^i) \in V_S$. Specifically, $S(v_S^i) = v_S^i$, and, for all v_i, only one *collection node* corresponds to v_i. The trajectory M of the *Mobile Sink* is composed of all nodes in V_S with the particular sequence. Figure 1 shows an example of *Mobile Sink* based data collection in large-scale wireless sensor networks.

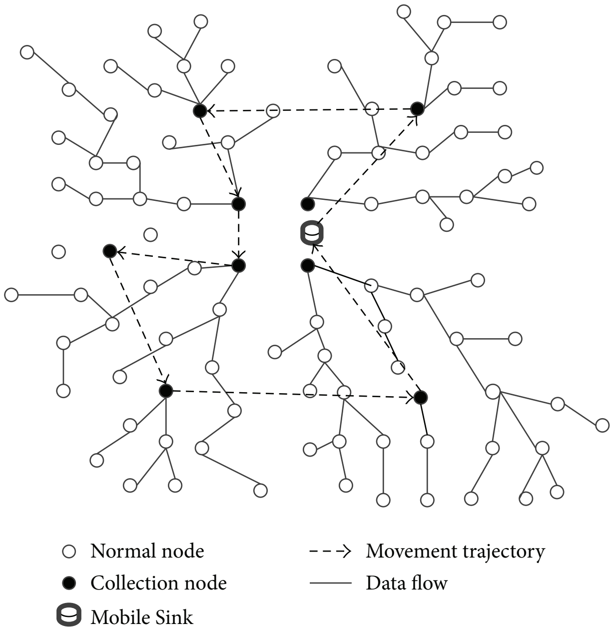


Figure 1: The *Mobile Sink* based data collection in large-scale wireless sensor networks.

This paper does not consider the issue of power control and assumes that all sensor nodes adopt the fixed transmitted power and received power; thus, the energy consumption is not related to the length between nodes. We adopt the energy consumption model described in [11, 22]. Within the single time period T, the energy consumption of any sensor nodes v_i is $E_i \approx a(p_r^i + p_t^i)$, where p_r^i presents the amount of received data and p_t^i presents the amount of data sent by v_i, and a is a constant, which presents the energy consumption for sending or receiving one bit of data. In the energy consumption model, the following relation holds: $p_t^i = p_r^i + q$, where q presents the total amount of the data generated in any node v_i within T. The amount of received data for all sensor nodes in the network within T is $\sum_{i=1}^{n} p_r^i = \sum_{i=1}^{n} h_i \cdot q$, where h_i presents the number of hops from the *normal node* v_i to $S(v_i)$. Then the total energy within the single time period T is

$$E_{\text{total}} = \sum_{i=1}^{n} E_i = \sum_{i=1}^{n} a\left(2h_i + 1\right) \cdot q. \qquad (1)$$

3.2. Problem Formulation. According to formula (1), we can reduce energy consumption by reducing the hops from the *normal nodes* to the corresponding *collection nodes*. We choose an ordered set of *collection nodes* V_S from the network and make each v_S^i, $i \in \{1, 2, \dots, |V_S|\}$, to be the root of the *collection tree* in order to reduce the total energy consumption of the network. Consider that $Q_i = E_{\text{total}} - E'_{\text{total}} > 0$ is energy reduction if we choose node v_i as the *collection node* within the single time period T, where E'_{total} is the total energy consumption within the single time period if we choose node v_i as the *collection node*. When putting v_i into V_S, Q_j may be changed for each node $v_j \in V \setminus V_S$, where $V \setminus V_S$ is the set of sensor nodes that are out of V_S.

Our objective is to design the trajectory M of the *Mobile Sink* to complete data collection for all sensor nodes and return back to the original position within the time period T with maximum total energy reduction. We call this problem *Maximizing Energy Reduction for Constrained Trajectory* (MERC), which can be formalized as follows:

decision variables:

$$x_{ij} = \begin{cases} 1, & \left(v_i, v_j\right) \text{ is on the } M, \\ 0, & \text{Otherwise}, \end{cases} \qquad i, j = 1, \dots, n, \qquad (2)$$

objective function:

$$\max \sum_{i=1}^{n-1} \sum_{j=2}^{n} Q_j x_{ij}, \tag{3}$$

constraints:

$$\sum_{j=2}^{n} x_{1j} = \sum_{i=2}^{n-1} x_{i1} = 1, \tag{4}$$

$$\sum_{i=1}^{n-1} x_{ik} = \sum_{j=2}^{n} x_{kj} \leq 1; \quad \forall k = 2, \ldots, n-1, \tag{5}$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^{n} l_{ij} x_{ij} \leq C \cdot T, \tag{6}$$

where (v_i, v_j) is the path from v_i to v_j and l_{ij} is the corresponding Euclidean distance.

The objective function (3) makes M to maximize the energy reduction. Constraint (4) presents that the *Mobile Sink* moves from the original position and returns to the original position finally. Constraint (5) presents that there is connectivity among the sensor nodes and each sensor node can only be accessed once in T. Constraint (6) presents the length constraint of M.

4. Constrained Trajectory Based on Maximizing Energy Reduction

4.1. Complexity Analysis of MERC Problem. First of all, we attempt to find a computable feasible algorithm for the *MERC* problem. Unfortunately, as the following lemma shows, it is NP-hard to find the optimal solution even Q_i for each node v_i is a constant, called static *MERC* conveniently.

Lemma 1. *The static MERC problem is NP-hard.*

Proof. We demonstrate that static MERC belongs to NP firstly. Given an instance of static MERC, we can check whether the length of M is no more than $C \cdot T$ and check whether the total reduction of energy is no less than k. This process can be ended up in polynomial time.

Next, we prove static MERC is NP-hard by giving a polynomial time reduction from the NP-hard Orienteering Problem [12], OP.

Instance of OP (denoted by $\mathscr{A}$): for a graph $G = (V, A)$, where $V = \{v_1, \ldots, v_n\}$ is the vertex set and A is the arc set. The nonnegative score S_i is associated with each vertex $v_i \in V$ and the travel time t_{ij} is associated with each arc $a_{ij} \in A$. For a positive real k, the question is whether Hamiltonian path $G'(\subset G)$ exists over a subset of V, such that the sum of score for each vertex in G' is no less than k, including preset start (v_1) and end (v_1) vertex with total travel time no more than $T_{\max}$.

We consider a corresponding instance of static MERC (denoted by $\mathscr{B}$): for a graph $G = (V, A)$, where $V = \{v_1, \ldots, v_n\}$ is the sensor node set and A is the arc set. The nonnegative reduction of energy Q_i is associated with each sensor node $v_i \in V$ and the Euclidean distance l_{ij} is associated with each arc $a_{ij} \in A$. For positive real k, C, and T, the question is whether Hamiltonian path (trajectory) $G'(\subset G)$ exists over a subset of V, such that the sum of reduction of energy for all sensor nodes in G' is no less than k, including preset start (v_1) and end (v_1) vertex with path length no more than $C \cdot T$.

This reduction from $\mathscr{A}$ to $\mathscr{B}$ ends in polynomial time. We can simply see that r is a solution of $\mathscr{A}$ if and only if r is a solution of $\mathscr{B}$. □

It is obvious that the MERC problem with dynamic Q_i is not easier than the static MERC problem. Thus, we have the following theorem.

Theorem 2. *The MERC problem is NP-hard.*

4.2. CTMER Design. Since the *MERC* problem is NP-hard, we turn our attention to develop an approximation algorithm. Given the time period T, the movement speed of the *Mobile Sink* C, the set of sensor nodes V, and a position vector $\mathbf{pos} = (\mathrm{pos}_0, \mathrm{pos}_1, \ldots, \mathrm{pos}_n)$, in which pos_0 is the original position of *Mobile Sink*, and pos_i, $i \in \{1, \ldots, n\}$, are position of sensor node v_i, the algorithm $\mathfrak{R}(C, T, V, \mathbf{pos})$ returns an ordered set V_S, which contains all selected *collection nodes*, and the length of trajectory $L = |M|$.

The design rationale is that CTMER first generates original *collection trees* based on the original position of the *Mobile Sink*, and the roots of the original *collection trees* are the initial *collection nodes*. And then CTMER iteratively selects the nodes with *maximum effective average energy reduction* in current topology state as the *collection nodes*. As illustrated in Algorithm 1, CTMER follows a greedy approach to solve *MERC* problem and designs the movement trajectory of the *Mobile Sink* before running.

CTMER uses the function **Initialize** to generate the original topology of the sensor network. As illustrated in Algorithm 2, the function **Initialize** will construct original *collection trees*, whose roots are located in the one hop range from the *Mobile Sink* at position pos_0. Many methods can construct *collection trees*, and we will give an instance in Section 4.3.

Considering each node $v_i \in V$ with energy reduction, Q_i, if the *Mobile Sink* visits v_i, v_i would be a new *collection node*, and the corresponding *collection tree* $\mathrm{Tree}(v_i)$ with root v_i can be structured. We call this operation as **Cut** for briefness. Obviously, when performing the **Cut** operation in position v_i, we have $Q_i = \mathrm{num}_i \cdot h_i$, where num_i is the number of nodes in the *collection tree* $\mathrm{Tree}(v_i)$ and h_i is the hop count from v_i to (v_i) before **Cut** operation. CTMER puts the node v_i into ordered set V_S iteratively based on *effective average energy reduction*, $Q_i/\Delta L_i$, until the next **Cut** operation will make $L > C \cdot T$. ΔL_i is the increased length of M, that is, the increased length of **TSP** circuit when the *Mobile Sink* visits v_i.

When performing a **Cut** operation, a new *collection tree* is structured, and the topology changed correspondingly. Without loss of generality, considering that v_j is selected as a new *collection node*, the impact of this **Cut** operation is restricted to the nodes located in the path from $S(v_j)$ to v_j

```
**Input**: time period $T$, movement speed of *Mobile Sink* $C$, the set of sensor nodes $V$, and position vector **pos**
**Output**: ordered set $V_S$, $L$
(1) $V_S \leftarrow \varnothing, L \leftarrow 0$
(2) **Initialize**($V$, $\mathrm{pos}_0$);
(3) **while** $L < C \cdot T$ **do**
(4)     **foreach** node $v_i$ in $V \setminus V_S$ **do** $\Delta L_i = \mathbf{TSP}(V_S \cup \{i\}) - \mathbf{TSP}(V_S)$;
(5)     $j = \arg\max_{i \in V \setminus V_S} \mathrm{num}_i \cdot h_i / \Delta L_i$;
(6)     $L \leftarrow L + \Delta L_j$;
(7)     **if** $L \le C \cdot T$ **then**
(8)         $V_S \leftarrow V_S \cup \{v_j\}$;
(9)         **Cut**($V$, $v_j$);
(10)    **else**
(11)        $L \leftarrow L - \Delta L_j$;
(12)    **end if**
(13) **end while**
(14) **return** ($V_S$, $L$);
```

Algorithm 1: CTMER.

```
**Input**: the set of sensor nodes $V$, the start position of the *Mobile Sink* $\mathrm{pos}_0$
**Output**: num, $h$
(1) Initialize collection trees based on $\mathrm{pos}_0$;
(2) **foreach** node $v_i$ in $V$ **do**
(3)     $h_i \leftarrow$ hop count from $S(v_i)$ to $v_i$;
(4)     $\mathrm{num}_i \leftarrow$ the number of nodes in Tree($v_i$);
(5) **end for**
(6) **return** (num, $h$);
```

Algorithm 2: **Initialize**.

```
**Input**: the set of sensor nodes $V$, the sensor node $v_j$
**Output**: num, $h$
(1) **foreach** node $v_i$ in the path from $S(v_j)$ to $v_j$ **do** $\mathrm{num}_i \leftarrow \mathrm{num}_i - \mathrm{num}_j$;
(2) **foreach** node $v_i$ in Tree($v_j$) with nondecreasing order of $\mathrm{num}_i$ **do** $h_i \leftarrow h_i - h_j$;
(3) **return** (num, $h$);
```

Algorithm 3: **Cut**.

and the nodes located in Tree (v_j). The **Cut** operation is illustrated in Algorithm 3.

4.3. An Instance of Collection Tree Construction. To design the movement trajectory of *Mobile Sink*, it is necessary to generate the original topology of the sensor network, which can be generated through many different methods. Even so, we give an instance of *collection tree* construction, which performs in Algorithm 2 (Line 1).

We consider the *Mobile Sink* is in arbitrary position pos_0 and use the example in Figure 2 to illustrate how the instance works. There are 22 sensor nodes and one *Mobile Sink*, and all sensor nodes have the same transmission range, which are presented as dashed annulus. The *Mobile Sink* sends a message to the neighbor sensor nodes (within one hop distance), and each node, which received the message, forwards the message to its neighbors again and sends an acknowledgement to the *Mobile Sink* by the reverse path. This process continues until the *Mobile Sink* received all nodes' acknowledgements. Finally, the original *collection trees* can be constructed, and the nodes within the *Mobile Sink*'s one hop distance are initialized as the *collection nodes*. Figure 3 shows the *collection trees* in this instance. By the above interactive process, it is not different to get num_i, h_i, and Q_i, which have been listed in Table 1, for each node v_i in network.

4.4. Computational Efficiency of CTMER. The computational efficiency of trajectory design algorithm is important in large-scale wireless sensor networks. We have the following theorem about the computational efficiency of CTMER.

Table 1: num_i, h_i, and Q_i for each node v_i in the example.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
num_i	2	4	1	1	2	1	4	1	2	1	4	1	2	1	7	4	2	2	1	1	1	7
h_i	1	1	2	2	2	3	0	1	1	2	0	1	1	2	0	1	1	2	3	2	2	0
Q_i	2	4	2	2	4	3	0	1	2	2	0	1	2	2	0	4	2	4	3	2	2	0

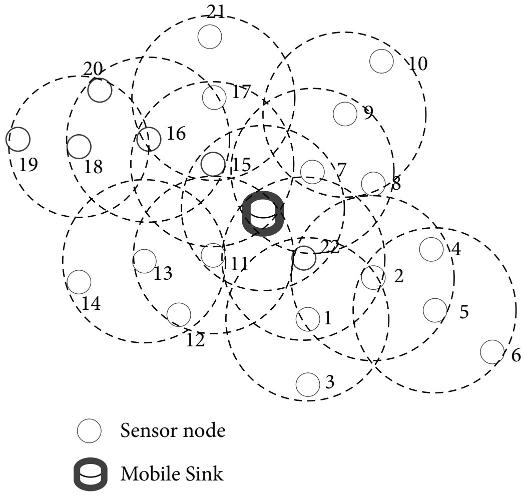


Figure 2: An instance of sensor network with one *Mobile Sink* and 22 sensor nodes.

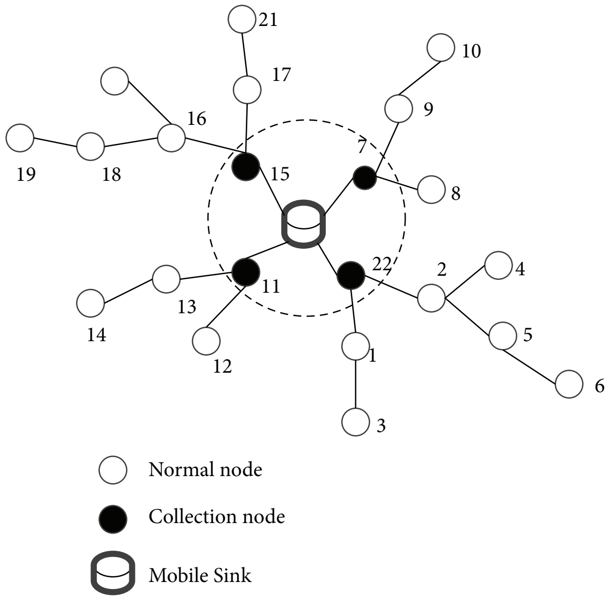


Figure 3: The collection trees initialized in our instance.

Table 2: Simulation parameters.

Parameter	Value
Network size (m^2)	500×500
Number of sensor nodes	1000
Initial energy of each sensor node (J)	2
Initial energy of the *Mobile Sink* (J)	1×10^6
Unit energy consumption (J/bit)	5×10^{-6}
Single time period (s)	600
Communication radius (m)	50
Speed of the *Mobile Sink* (m/s)	2
Packet size (bit)	500
Packet generation ratio (packet/minute)	1
Original position of the *Mobile Sink*	(100, 100)

Theorem 3. *CTMER is computationally efficient.*

Proof. The function **Initialize** (Line 2) takes $O(n^2)$ time. Since CTMER performs cut operation for each node in V at most once, the while-loop (Lines 3–13) takes $O(n)$ time. CTMER calculates the TSP circuit for each node in $V \setminus V_S$. Since we use the Euclidean distance, the **TSP** circuit can be achieved in $O(n^2)$ within 2 approximations. Thus, the for-loop (Line 4) takes $O(n^3)$ time. Finding the maximum effective average energy reduction (Line 5) takes $O(n)$ time. The **Cut** operation (Line 9) takes $O(n)$ time. Hence, the running time of the whole CTMER is bounded by $O(n^4)$. □

Note that the running time of CTMER, $O(n^4)$, is very conservative since the number of *collection nodes* is much less than n in practice. Moreover, the accurate complexity of the for-loop (Line 4) is $\sum_{i=0}^{n} i^2 = n(n+1)(2n+1)/6$ since CTMER only calculates the **TSP** circuit containing $V_S \cup \{i\}$ for the nodes in $V \setminus V_S$.

5. Performance Evaluation

5.1. Simulation Setup. We implement CTMER on the MATLAB platform and compare our protocol to RD-VT (with a variable BS track), QDD (with one *Mobile Sink*), and LBDD (with one *Mobile Sink* using the progressive-footprint-chaining strategy) in order to investigate the performance of CTMER. We list the default parameter setting in Table 2. Each measurement is averaged over 200 instances.

Since the object of CTMER is maximizing the total energy reduction, we focus on the following four aspects, which are key metrics in large-scale WSNs:

(1) Energy consumption: this represents the sum of the energy consumption of all sensor nodes in wireless sensor network within first 600 s with fixed message generation ratio. We use the energy consumption model described in Section 3 and investigate the performance with different number of sensor nodes and different speed of the *Mobile Sink*. The LBDD and QDD follow query mode other than collecting data periodically; thus, we set the number of queries as the

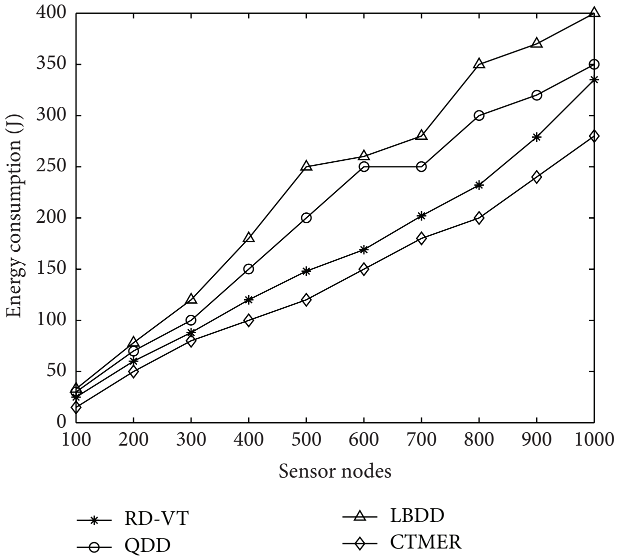


Figure 4: Energy consumption with different number of sensor nodes.

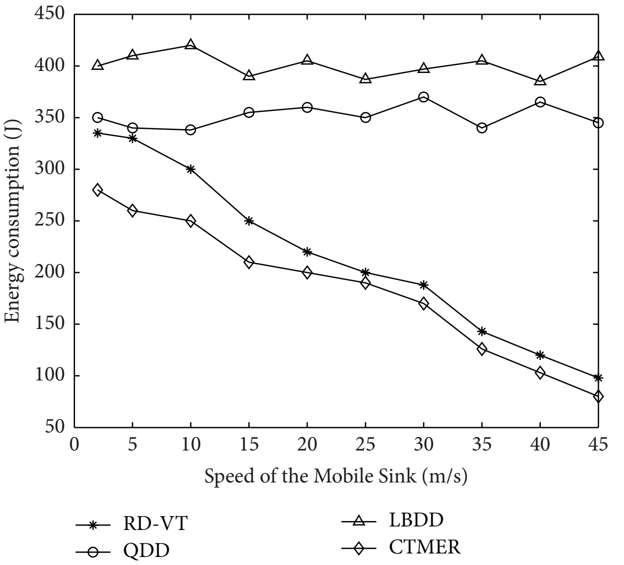


Figure 5: Energy consumption with different speed of the *Mobile Sink*.

number of sensor nodes. This means that the *Mobile Sink* queries data once for each sensor node within first 600 s.

(2) Network lifetime: this represents the elapsed time from the start to the end of the first node energy depletion.

(3) Total path length: this represents the total hops for all sensor nodes to forward the data to the *Mobile Sink*. We do not test the total path length of LBDD and QDD, since there is no actual routing tree in these two protocols.

(4) Numbers of Cut operations: this represents the number of Cut operations when we use CTMER algorithm to optimize the energy. We vary the speed of the *Mobile Sink* to investigate the impact on the number of Cut operations.

5.2. Analysis of Results. We first investigate the impact on the energy consumption of the sensor network with the increasing number of sensor nodes. As can be seen from Figure 4, the energy consumption is increasing with the increasing number of nodes for all four protocols. RD-VT minimizes the total Euclidean distance of edges in the routing trees other than the total hop count from *sensor nodes* to *collection nodes*; thus, it does not perform good enough comparing with CTMER. The energy consumption of QDD and LBDD is higher than both CTMER and RD-VT because a large number of queries from the *Mobile Sink* and data from *rendezvous points* need multiple hop forwarding. For LBDD, the sources need to forward the data to the first inline-node, which may be far away from the sources. CTMER uses the Cut operations to optimize the total path length of the collection trees, which makes significant impact on energy consumption in our energy consumption model, and performs best in all compared protocols.

Figure 5 shows the impact on the energy consumption with increasing speed of the *Mobile Sink*. In CTMER, the increasing moving speed means the increasing length of the trajectory when the single time period is fixed and incurs more Cut operations; thus, the energy consumption of CTMER reduces gradually. With the similar reason, RD-VT can choose more *rendezvous points* in the Steiner Minimum Tree. However, the number of *rendezvous points or inline-nodes* is independent with the speed of the *Mobile Sink* in LBDD and QDD; thus, the energy consumption of them has no distinct change.

We then test the network lifetime of the protocols. We can see from Figure 6 that the network lifetime of all protocols decreases with the increasing number of sensor nodes. In hierarchical structure based method, the network lifetime is significantly relevant to the overhead of the *collection nodes*. In CTMER and RD-VT, the overhead of *collection nodes* is determined by the number of sensor nodes in *collection trees* since the packet generation ratio is fixed. With the increasing sensor nodes, there are more sensor nodes in *collection trees* in average, and the overhead of *collection nodes* increases accordingly in both CTMER and RD-VT. The performance of CTMER and RD-VT is very closed in aspect of network lifetime. However, the overhead of *rendezvous points* in high lever will increase significantly in QDD, and the network lifetime decreases dramatically with the increasing scale of network.

Figure 7 shows the impact on the network lifetime with the increasing speeds of the *Mobile Sink*. As the *collection tree* based protocols, the network lifetime of both CTMER and RD-VT increases dramatically because there are more *collection nodes* to share the overhead. On the contrary, there is almost no influence on QDD and LBDD. This is because

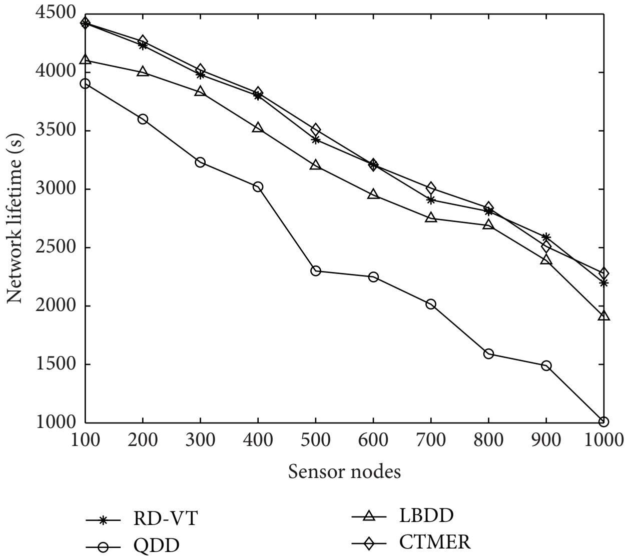


Figure 6: Network lifetime with different number of sensor nodes.

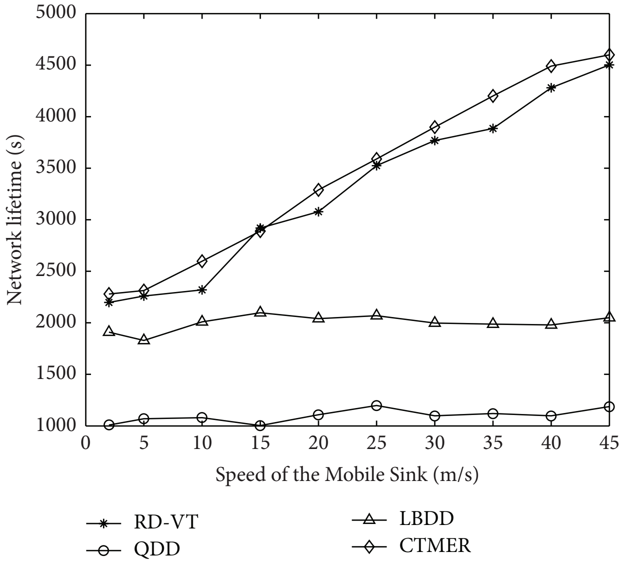


Figure 7: Network lifetime with different speed of the *Mobile Sink*.

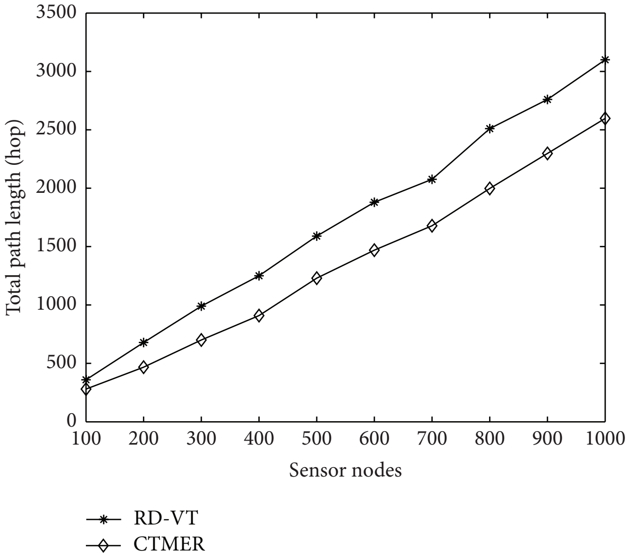


Figure 8: Total path length with different number of sensor nodes.

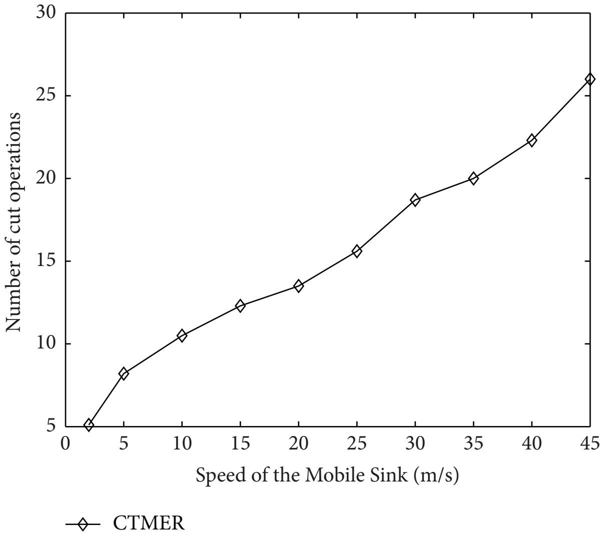


Figure 9: Number of Cut operations with different speed of the *Mobile Sink*.

the *rendezvous points* in QDD and LBDD are selected based on the structure and scale of network and are irrelevant to the speed of the *Mobile Sink*.

Figure 8 shows the total path length of CTMER and RD-VT with the increasing sensor nodes. Recall that the total path length is the total hop count from all sensor nodes to the corresponding *collection nodes*, and it determines the total energy consumption in single time period based on the energy consumption model used in this paper. The total path length of both CTMER and RD-VT increases with the increasing network scale. We can see that CTMER is superior to RD-VT in aspect of total path length. This is because CTMER and RD-VT use different method to choose the *collection nodes* to achieve the different objectives. CTMER aims to optimize the total hop count while RD-VT minimizes the total Euclidean distance of edges in the routing trees.

CTMER chooses the *collection nodes* iteratively until the trajectory length of the *Mobile Sink* does not meet the constraint. Thus, CTMER can do more Cut operations when the limit length of trajectory becomes longer. We can see from Figure 9 that the number of Cut operations increases dramatically from 5.1 to 26.2 with the speed of the *Mobile Sink* since the *Mobile Sink* can move far away in single time period.

6. Conclusion

In this paper, we have presented the system model for designing constrained trajectory in large-scale wireless sensor networks under our energy consumption model and formulated the problem as the MERC problem. We designed an approximation algorithm (CTMER), which follows the greedy approach to design the movement trajectory of the *Mobile Sink* through maximizing the *effective average energy reduction*. CTMER uses Cut operations to choose the *collection nodes* iteratively until the trajectory length of the *Mobile Sink* does not meet the constraint. Through both rigid theoretical analysis and extensive simulations, we demonstrated that our algorithm achieves high computation efficiency and is superior to other *Mobile Sink* based data collection methods in aspects of energy consumption and network lifetime.

Conflict of Interests

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

Acknowledgments

This work is sponsored in part by NSFC Grant (nos. 61472193, 61472192, 61373139, 61300240, and 61472283), the Natural Science Foundation of Jiangsu Province (nos. BK20141429, BK20130852, and BK20151511), Scientific and Technological Support Project (Society) of Jiangsu Province (no. BE2013666), CCF-Tencent Open Research Fund (AGR20150107), China Postdoctoral Science Foundation (no. 2014M562662), Jiangsu Postdoctoral Science Foundation (no. 1402223C), Project of Natural Science Research of Jiangsu University (no. 14KJB520027), and Scientific and Technological Support Project (Society) of Lianyungang (no. SH1306).

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] V. Potdar, A. Sharif, and E. Chang, “Wireless sensor networks: a survey,” in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA ’09)*, pp. 636–641, May 2009.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] A. Chakrabarti, A. Sabharwal, and B. Aazhang, “Communication power optimization in a sensor network with a path-constrained mobile observer,” *ACM Transactions on Sensor Networks*, vol. 2, no. 3, pp. 297–324, 2006.

[5] L. Song and D. Hatzinakos, “Architecture of wireless sensor networks with mobile sinks: sparsely deployed sensors,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 1826–1836, 2007.

[6] S. Gao, H. Zhang, and S. K. Das, “Efficient data collection in wireless sensor networks with path-constrained mobile sinks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 592–608, 2011.

[7] P. Baruah, R. Urgaonkar, and B. Krishnamachari, “Learning-enforced time domain routing to mobile sinks in wireless sensor fields,” in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN ’04)*, pp. 525–532, IEEE, New York, NY, USA, November 2004.

[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. 231–239, ACM, Hong Kong, May 2008.

[9] E. B. Hamida and G. Chelius, “A line-based data dissemination protocol for wireless sensor networks with mobile sink,” in *Proceedings of the IEEE International Conference on Communications (ICC ’08)*, pp. 2201–2205, Beijing, China, May 2008.

[10] M. Z. Hameed and K. Young-Bae, “A quadtree-based data dissemination protocol for wireless sensor networks with mobile sinks,” in *Proceedings of the 11th International Conference Personal Wireless Communications*, pp. 447–456, September 2006.

[11] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, “Exploiting sink mobility for maximizing sensor networks lifetime,” in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, p. 287, IEEE, January 2005.

[12] B. L. Goldn, L. Levy, and R. Vohra, “The orienteering problem,” *Naval Research Logistics*, vol. 34, no. 3, pp. 307–318, 1987.

[13] L. He, J. P. Pan, and J. D. Xu, “A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1308–1320, 2013.

[14] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: modeling a three-tier architecture for sparse sensor networks,” in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA ’03)*, pp. 215–233, IEEE Press, New York, NY, USA, October 2003.

[15] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, “Robomote: enabling mobility in sensor networks,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 404–409, ACM/IEEE Press, New York, NY, USA, April 2005.

[16] D. Lymberopoulos and A. Savvides, “XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN ’05)*, pp. 449–454, ACM/IEEE Press, April 2005.

[17] 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.

[18] P. Richard, G. Jason, K. Aman et al., “Networked info mechanical systems: a mobile embedded networked sensor platform,” in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN ’05)*, pp. 376–381, ACM/IEEE Press, Los Angeles, Calif, USA, August 2005.

[19] X.-W. Zhang, H.-P. Dai, L.-J. Xu, and G.-H. Chen, “Mobility-assisted data gathering strategies in WSNs,” *Journal of Software*, vol. 24, no. 2, pp. 198–214, 2013 (Chinese).

[20] N. Mitton, E. Natalizio, and R. Wolhuter, “Beacon-less mobility assisted energy efficient georouting in energy harvesting actuator and sensor networks,” in *Proceedings of the 12th International Conference on Ad-Hoc, Mobile, and Wireless Network*, pp. 281–292, Wrocław, Poland, July 2013.

[21] F. Tashtarian, M. H. Y. Moghaddam, K. Sohraby, and S. Effati, “ODT: optimal deadline-based trajectory for mobile sinks in WSN: a decision tree and dynamic programming approach,” *Computer Networks*, vol. 77, pp. 128–143, 2015.

[22] S. Gao, H.-K. Zhang, and H.-S. Xu, “Efficient data gathering approach in sensor networks with path-fixed sinks,” *Journal of Software*, vol. 21, no. 1, pp. 147–162, 2010 (Chinese).

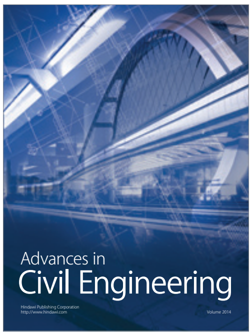

Journal of
Engineering

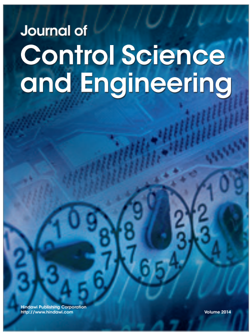

The Scientific World Journal


International Journal of
Rotating Machinery


Journal of
Sensors

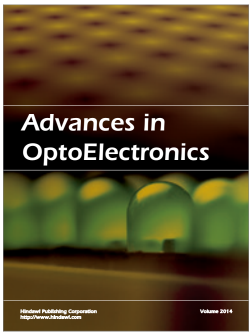

International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

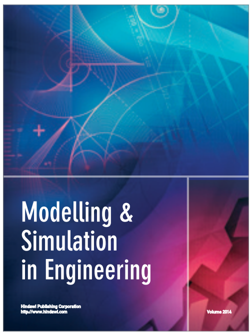

Journal of
Robotics


Journal of
Electrical and Computer Engineering

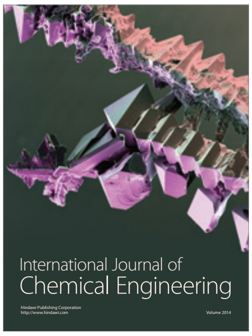

Advances in
OptoElectronics

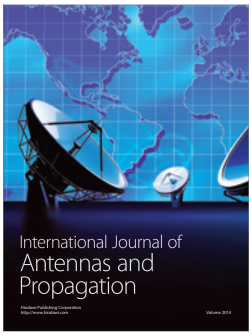

VLSI Design


International Journal of
Navigation and Observation

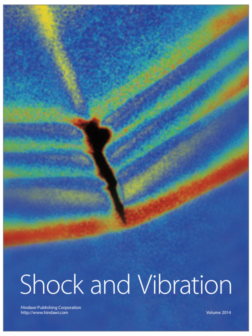

Modelling & Simulation in Engineering

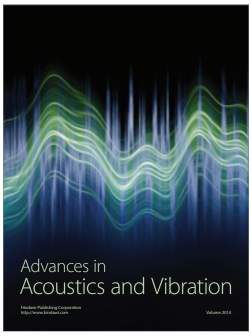

International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation


Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration