www.ijsetr.com

International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.05,Issue.01,
January-2016,
Pages:0008-0018

Implementation of Wireless Sensor Networks Based on Minimizing Movement for Target Coverage and Network Connectivity in Mobile Sensor Networks

D. PRASAD
Professor, Dept of ECE, Ramanandatirtha Engineering College, Nalgonda, TS, India.

Abstract: Coverage of interest points and network connectivity are two main challenging and practically important issues of Wireless Sensor Networks (WSNs). Although many studies have exploited the mobility of sensors to improve the quality of coverage and connectivity, little attention has been paid to the minimization of sensors' movement, which often consumes the majority of the limited energy of sensors and thus shortens the network lifetime significantly. To fill in this gap, this paper addresses the challenges of the Mobile Sensor Deployment (MSD) problem and investigates how to deploy mobile sensors with minimum movement to form a WSN that provides both target coverage and network connectivity. To this end, the MSD problem is decomposed into two sub-problems: the Target COVerage (TCOV) problem and the Network CONnectivity (NCON) problem. We then solve TCOV and NCON one by one and combine their solutions to address the MSD problem. The NP-hardness of TCOV is proved. For a special case of TCOV where targets disperse from each other farther than double of the coverage radius, an exact algorithm based on the Hungarian method is proposed to find the optimal solution. For general cases of TCOV, two heuristic algorithms, i.e., the Basic algorithm based on clique partition and the TV-Greedy algorithm based on Voronoi partition of the deployment region, are proposed to reduce the total movement distance of sensors. For NCON, an efficient solution based on the Steiner minimum tree with constrained edge length is proposed. The combination of the solutions to TCOV and NCON, as demonstrated by extensive simulation experiments, offers a promising solution to the original MSD problem that balances the load of different sensors and prolongs the network lifetime consequently.

Keywords: Wireless Sensor Networks, Target Coverage, Connectivity, Mobile Sensors, Energy Consumption.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are currently used in a wide range of applications including environmental monitoring [1] and object tracking [2]. Target coverage and connectivity are two main challenging and practically important issues of WSNs. Target coverage aims to cover a set of specified points of interest in the deployment region of a WSN. It characterizes the monitoring quality of the network [3]. Connectivity is necessary for sensors in a WSN to collect data and report data to the sink node. However, WSNs formed by randomly distributed wireless sensor nodes often cannot provide satisfactory coverage quality and cannot guarantee the connectivity of the network. In recent years, sensor mobility has been exploited to improve the coverage quality and connectivity in randomly deployed WSNs by relocating some mobile sensors to new Positions to enhance the coverage quality and the connectivity of the network [4], [5], [6], [7], [8]. In this paper, we address a practically important problem of minimizing sensors' movement to achieve both target coverage and network connectivity in mobile sensor networks. As sensors are usually powered by energy limited batteries and thus severely power-constrained, energy consumption should be the top consideration in mobile sensor networks. Specially, movement of sensors should be minimized to prolong the network lifetime because sensor movement consumes much more energy than sensing and communication do [6], [9]. However, most of the existing studies aimed at improving the quality of target coverage, e.g., detecting targets with high detection probability, lowering false alarm rate and detection delay. Little attention has been paid to minimizing sensor movement. To fill in this gap, this study focuses on moving sensors to cover discrete targets and form a connected network with minimum movement and energy consumption. To this end, we first formulate the Mobile Sensor Deployment (MSD) problem with the aim of deploying mobile sensors to provide target coverage and network connectivity with minimum movement. The MSD problem is then decomposed into two sub-problems: Target COVerage (TCOV) and Network CONnectivity (NCON). Combining the solutions to the two sub-problems, we achieve an efficient solution to the MSD problem.The main contributions of this paper are summarized as follows:

- We prove the NP-hardness of the TCOV problem. For a special case of TCOV in which targets disperse from each other by more than double of the coverage radius, an exact algorithm based on the extended Hungarian method is proposed to find the optimal solution to TCOV.

- For the general case of TCOV, two heuristic algorithms are proposed: the Basic algorithm based on clique partition, and the TV-Greedy algorithm based on Voronoi partition diagram of target points. The Basic algorithm reduces the total movement distance by minimizing the number of sensors to be moved. The TV-Greedy algorithm minimizes the total movement distance by grouping and dispatching sensors according to their proximity to targets in the Voronoi diagram.
- For the NCON problem, first an edge length constrained Steiner tree is constructed to determine the Steiner points that are needed to connect the coverage sensors and the sink, then the extended Hungarian method is used to find the optimal sensors to move to these points.
- Extensive simulation experiments are conducted to evaluate the performance of the proposed algorithms. The results demonstrate that the combination of the solutions to TCOV and NCON offers a promising solution to the original MSD problem, as well as balances the load of different sensors and prolongs the network lifetime consequently.

II. RELATED WORK

With the emergence of mobile sensors, extensive researches have been promoted on target coverage of WSNs. According to different application scenarios, the existing studies can be classified into three categories: (1) route patrol for collecting data from fixed targets [10], [11], [12], (2) detection of mobile targets [4], [5], [13], and (3) target coverage in dynamic environments [14], [15]. In these studies, mobile sensors move actively to improve the surveillance quality, but the optimization of sensor movement is not explicitly considered. Reactive mobility is exploited to improve the quality of target detection in [6], but the movement of sensors is not considered as the primary optimization objective. In [7] mobile sensors are scheduled to replace failed static sensors in order to guarantee coverage ratio with minimum movement distance. But each sensor concerned in [7] can cover only one target and the maximum moving distance for each mobile sensor is limited. In [16], an optimal velocity schedule is proposed to minimize energy consumption in movement when the road condition is uniform. Many research efforts have also been made to improve the area coverage with mobile sensors with the aim of maximizing the covered area. In [17], Voronoi diagrams are used to detect coverage holes. After that, sensors are dispatched to cover the detected holes. As a result, the area coverage ratio is improved. Further, a multiplicatives weighted Voronoi diagram is used to discover the coverage holes corresponding to different sensors with different sensing ranges [18].

However, Voronoi diagram to discover the coverage holes corresponding to different sensors with different sensing ranges. Voronoi diagrams in these studies are constructed according to the position of mobile sensors, and thus need to be recomputed after each round of sensor movement. In [19], mobile sensors are used to improve energy efficiency of sensors in area coverage. In this work, when destinations have been determined, mobile sensors are designed to move along the shortest path to minimize the energy consumption. Given designated destinations, k-coverage is studied in [20]. In this work, a competition scheme is proposed to minimize energy consumption in movement. Recently, parameterized algorithms were exploited to find maxlifetime target coverage [21] and min-power multicast paths [22] in WSNs. In these studies, destinations of mobile sensors are given in advance, and the energy efficiency is considered in the path finding process. Mobility of sensors could also be exploited to enhance network connectivity after the coverage stage is completed. In [23], a triangular deployment strategy is proposed to dispatch sensors to connect the network after deploying mobile routers to maximize the coverage area. In the proposed strategy, sensors move along the shortest path to the corresponding triangular vertices in order to save energy. In [24], the authors considered a hybrid network consisting of both static and mobile sensors. It first divides the static sensors into groups as large as possible, and then seeks the minimum number of mobile sensors to connect these static sensor groups.

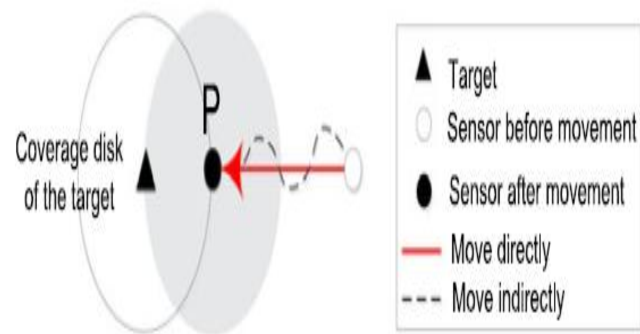


Fig.1. A sensor should move along the straight line between its initial position and the target to minimize the movement distance. In this example, the destination of the mobile sensor is P.

In [25], a sensor node relocation approach is proposed to maintain connectivity between a region of interest and a center of interest outside the deployment region where a particular event happens. The originality of this study and differences from the existing work include. (1) In this work, sensors move reactively and each sensor can cover more than one target, which is more general in practice, but also makes the problem more complicated. (2) The Voronoi diagram of targets is adopted to find the nearest sensor, which avoids blind competition among mobile sensors. Besides, because our solution generates the Voronoi diagram according to the position of targets, it does not require re-computation of the Voronoi diagram as the targets are static. This contributes to the lower complexity of the proposed solution. (3) Destinations of mobile sensors are unknown, which should be computed by our algorithms. When mobile sensors move to these destinations, both target coverage and network connectivity are satisfied. (4) In order to investigate the impact of network parameters on the performance of our algorithms, analyses and evaluations are given according to the simulation experiment results, which provides a reference for practical engineering and theoretical basis for the design of mobile sensor networks.

III. SYSTEM MODEL AND PROBLEM STATEMENT

A. System Model

In the system model addressed in this study, there are m targets $T = \{t_1, \ldots, t_m\}$ with known locations to be covered, and n mobile sensors $S = \{s_1, \ldots, s_n\}$ randomly deployed in the task area. The system model works as follows:

- Every mobile sensor knows its own position via a mounted GPS unit or a localization service in the network. We also assume that, there is a control center, e.g., a sink, which collects sensors' location information and broadcasts movement orders to mobile sensors.
- The task area is free of obstacles against movement. For the case with obstacles, a sensor is able to choose an appropriate shortest path to the destination to bypass the obstacles on the way. In this work, we focus on determining WHICH sensors should move and WHERE they should move to in order to guarantee both target coverage and network connectivity.
- **Network Model:** Disk model [26] is adopted for both sensing and communication of sensors with the sensing radius r_s and the communication radius r_c, respectively. Each target can be covered by more than one sensor, and each sensor can cover more than one target. A target is said covered if and only if there is at least one sensor in the disk of radius r_s centered at the target. The disk is defined as the target's coverage disk, and the circle of the coverage disk is called the target's coverage circle.
- **Mobility Model:** The free mobility model [6] is adopted. In this model, sensors are able to move continuously in any direction and stop anywhere. The distance that a sensor moves is used to present the sensor's energy consumption incurred in the movement. The movement distance of sensor s to cover target t is $dist(s, t) - r_s$, where dist(s, t) is the euclidean distance between s and t. Similarly, the movement distance of sensor s_i to connect with sensor s_j is $dist(s_i, s_j) - r_c$, where $\mathrm{dist}(s_i, s_j)$ is the distance between s_i and s_j. In the obstacle-free scenario, in order to minimize the movement distance of a sensor to a target, the sensor should move along the straight line from its initial position to the target until it reaches the target's coverage circle.

B. Problem Statement

1. Problem Definition: With the aforementioned system model, the formal definition of the MSD problem can be given as follows.

Definition 1: Mobile Sensor Deployment (MSD) problem: given m targets with known locations and n mobile sensors deployed randomly in the task area, the MSD problem seeks the minimum movement of mobile sensors such that the following objectives are achieved after mobile sensors reach their new positions:

- Every target is covered by at least one mobile sensor.
- The network formed by all the moved sensors is connected.

The MSD problem concerns two issues, namely target coverage and network connectivity. Thus, we divide it into two sub-problems and conquer them one by one. First, we focus on deploying mobile sensors to cover targets with minimum movement. These mobile sensors are called coverage sensors. Next, we deploy the rest sensors to provide connectivity between coverage sensors and the sink. The definitions of the two sub-problems are given below.

Definition 2: Target COVerage (TCOV) problem: given m targets with known locations and n mobile sensors deployed randomly in the task area, move sensors to new positions such that all the targets are covered and the total movement of sensors is minimized.

Definition 3: Network CONnectivity (NCON) problem: given a sink, the set of coverage sensors, and the rest mobile sensors after the TCOV problem is solved, NCON seeks the deployment of the rest mobile sensors to connect coverage sensors and the sink with minimum movement.

Theorem 1: In TCOV with free mobility model, in order to minimize the movement distance of mobile sensors, the number of potential positions to which sensors can move is finite.

Proof: Assume that there are m targets and n sensors in the network. We first consider the simple case in which one sensor covers exactly one target. Under the free mobility model, if one sensor wants to cover a target, it should move along the straight line connecting the sensor and the target and stop at the intersection of the line and the target's coverage circle, as shown in Fig. 1. As there are totally m targets and for each target there is exactly one optimal destination, the total number of potential destination points for the sensor is m.

We then consider the general case in which a sensor can cover more than one target. In this case, because several targets could be covered by one sensor simultaneously, their coverage disks intersect with each other. Denote the intersection part of these coverage disks as I. As the coverage disks are convex, the intersection I is also convex [27]. For a sensor outside of I, there exists a unique point closest to the sensor on the boundary of I. For instance, as shown in Fig. 2, the coverage disks of targets A, B, and C intersect with each other. For sensor S, P is the closest point to S in I and thus should be the destination of S if S is dispatched to cover the three targets. Similarly, P' is the unique closest point to S' in I. For k $(1 \leq k \leq m)$, the maximum number of k target combinations that could be covered by a single sensor is C_m^k. Thus the possible positions for a sensor to move to cover k targets is at most C_m^k. The total number of potential positions for a sensor to cover multiple targets is thus bounded by $\sum_{k=2}^{m} C_m^k$. From the above analysis, to cover m targets, the number of potential positions a sensor can move to is upper bounded by $1 + m + \sum_{k=2}^{m} C_m^k = 2^m$, where the term 1 corresponds to the case that the sensor stays at its original position. As there are n sensors, the total number of potential positions the mobile sensors can move to is at most $n*2^m$, which is finite as n and m are both finite. According to Theorem 1, when a sensor moves to one of its potential positions, a subset of targets are covered. Thus each potential position of a sensor corresponds to a subset of targets. For all the n sensors, the total number of potential positions is limited by $n*2^m$, which is finite. This conclusion is critical to help prove the hardness of the TCOV problem in the next section.

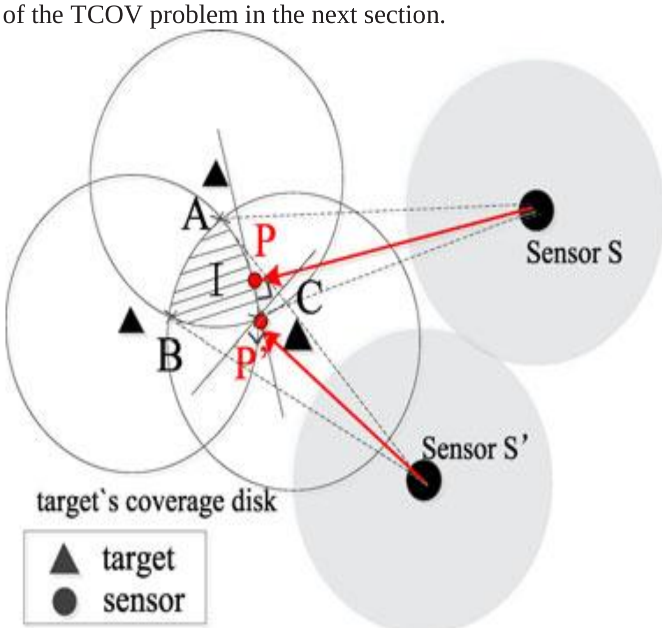


Fig. 2. Targets A, B, and C could be simultaneously covered by a single sensor (S or S0). Among all the points in I, P is closest to S, and thus it should be the destination position of S if S is despatched to cover the three targets. Similarly, P0 should be the destination position of S0 if S0 is despatched to cover the targets.

2. Hardness of the Problem: To show the hardness of the TCOV problem, we define a special case of TCOV, namely TCOV*, and prove that it is NP-hard. This naturally induces the NP-hardness of the original TCOV problem. The TCOV* problem is a special case of TCOV. In TCOV*, all the mobile sensors are initially deployed at the same location, which means that sensors start to move at the same point. If there exists a solution to TCOV, then the corresponding TCOV* problem will be solved by deploying mobile sensors at the same initial position, but the converse does not hold.

Theorem 2: The TCOV* problem is NP-hard.

Proof: Denote the power set of T (i.e., the set of all the targets) by P(T). Recall that there are totally n*2m potential positions to which the n mobile sensors move in the TCOV* problem. Each potential position corresponds to a subset of T, i.e., an element in P(T). For each potential position, we assign a weight W to its corresponding element in P(T), which is defined as the movement distance between the mobile sensor and that potential position. Then the decision version of the TCOV* problem can be transformed into the following set cover problem: given the universal set of targets T and a finite number (no more than n*2m) of weighted subsets of T whose union comprises the universe, determine whether there are some subsets whose total weight is less than or equal to W, such that the union of these subsets contains all the elements in T? The decision version of TCOV* is equivalent to that of the weighted set cover problem [28], which is NP complete. Therefore, TCOV* is NP-hard.

IV. SOLUTIONS TO THE TCOV PROBLEM

Although the TCOV problem is NP-hard, there exists a special case that can be solved in polynomial time. In this section, we first analyze a special case of TCOV and transform it into an assignment problem [29] and find the optimal solution. Furthermore, we propose two heuristic algorithms to solve the TCOV problem in the general case: the Basic algorithm based on clique partition, and the TV-Greedy algorithm based on the Voronoi partition diagram of targets.

A. Exact Solutions to a Special Case of TCOV

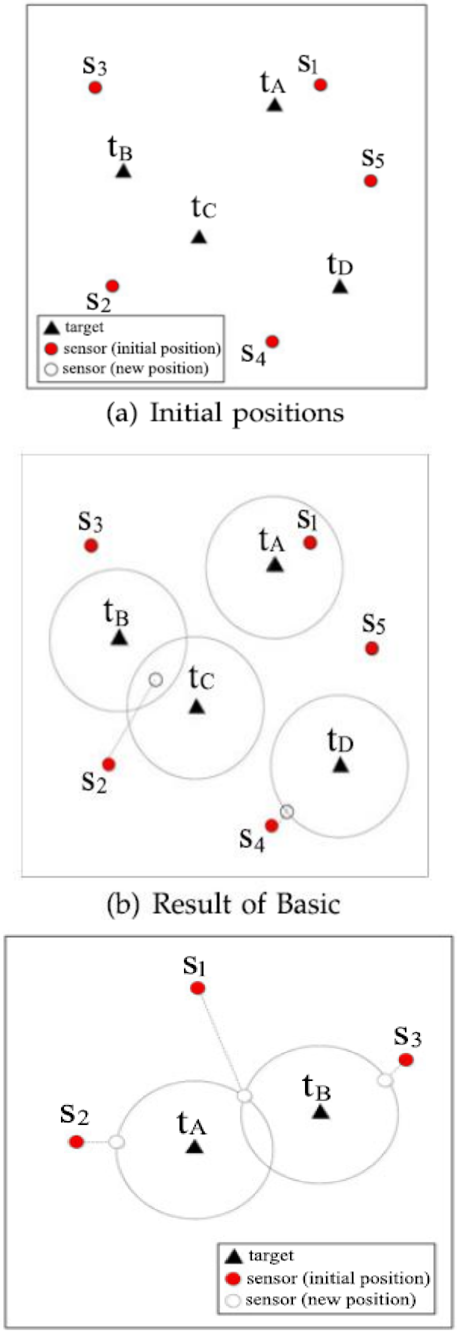
For a special case of TCOV in which the distance between any pair of targets is greater than 2rs, an exact solution based on the extended Hungarian algorithm is proposed in our previous work [30]. In this special case, as targets disperse from each other by more than double of the coverage radius, each sensor can cover at most one target. Thus, different targets need to be covered by different mobile sensors. The TCOV problem in this scenario could be transferred to the assignment problem [29] that is to assign exactly one agent to each task in such a way that the total cost of the assignment is minimized. However, in the traditional assignment problem the number of agents equals the number of tasks (n = m), while in our TCOV problem the number of sensors is usually larger than the number of targets, i.e., n > m. To deal with this issue, we extended the Hungarian algorithm proposed in [31] by extending the cost matrix to an n*n matrix as follows:

$$[c_{i,j}]_{n\times n} = \left(\begin{array}{ccc|ccc} c_{1,1} & \cdots & c_{1,m} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ c_{n,1} & \cdots & c_{n,m} & 0 & \cdots & 0 \end{array}\right), \quad \textbf{(1)}$$

where $c_{i,j}\ (1 \leq i \leq n, 1 \leq j \leq m)$ is set as the movement distance of moving sensor s_i to cover target t_j, i.e.,

$$c_{i,j} = \begin{cases} dist(s_i, t_j) - r_s & \text{if } dist(s_i, t_j) > r_s, \\ 0 & \text{otherwise,} \end{cases} \quad \textbf{(2)}$$

where dist(s_i; t_j) is the euclidean distance between s_i and t_j. With this extended cost matrix, the optimal solution to TCOV in the special case could be found in polynomial time by using the Hungarian algorithm [31]. More details on the extended Hungarian algorithm can be found in our previous work [30].



(a) Initial positions

(b) Result of Basic

(c) Suboptimality of Basic

Fig. 3. Illustration of the Basic algorithm: (a) Initial positions of targets and sensors; (b) The results of the Basic algorithm, in which two sensors need to move; (c) Suboptimality of the Basic algorithm: moving least sensors may induce longer total movement distance.

B. Heuristic Solutions to the General Case of TCOV

For the general case of the TCOV problem, we propose two heuristic algorithms to find near optimal solutions.

The Basic Algorithm: A simple heuristic to minimize the movement distance of sensors is to minimize the number of sensors that need to move. Actually, after the sensors are deployed, some targets may have already been covered. Denote the set of targets that have already been covered by $T_{initcov}$, and denote the set of uncovered targets by $T_{needcov}$. Then we have $T_{needcov} = T \setminus T_{inticov}$. In order to minimize the number of mobile sensors that need to move, we first construct a graph of targets representing whether targets can be simultaneously covered, then find the destinations of mobile sensors by using clique partition. The graph is constructed as follows. For every target in $T_{needcov}$, there is a vertex in the graph. There is an edge between two vertices if and only if the corresponding targets could be simultaneously covered by the same sensor. After the graph is constructed, we find a minimum clique partition of the constructed graph. Each partitioned clique represents a subset of targets that can be covered by the same sensor. Thus, for targets belonging the same clique, we need to dispatch only one mobile sensor to cover them. With this method, the number of mobile sensors that need to move is minimized. After the clique partition is obtained, the extended Hungarian algorithm is used to determine which sensor should be dispatched to cover the targets in each clique.

Algorithm 1. The Basic Algorithm

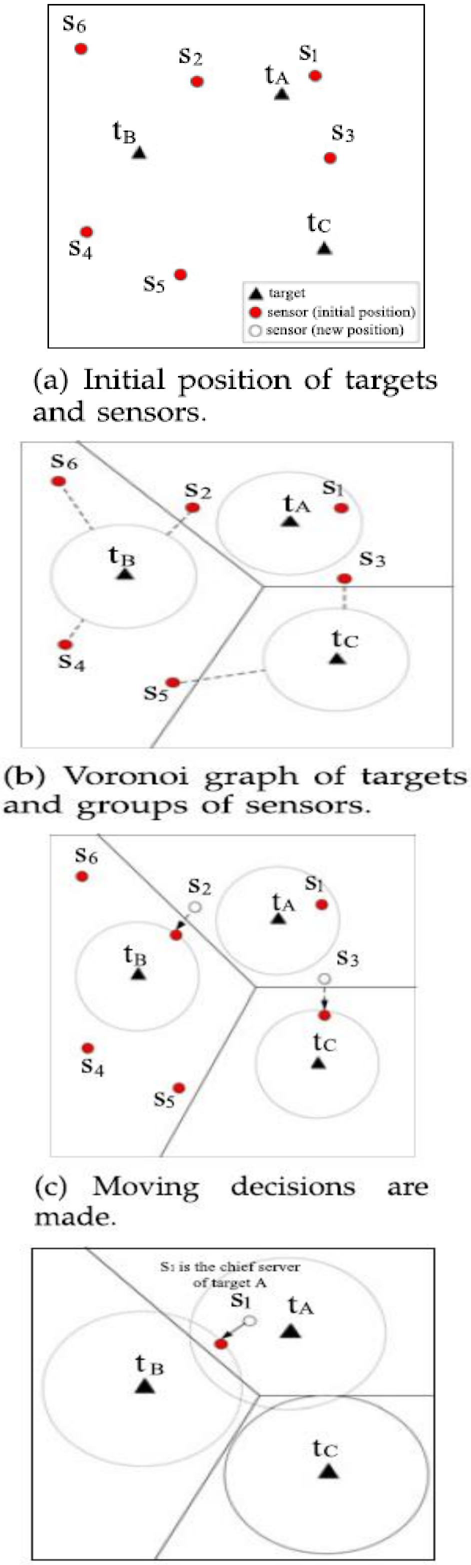
Input: $T = t_1, t_2, \ldots, t_m$;// Positions of targets
$S = s_1, s_2, \ldots, s_n$;// Initial positions of sensors
r_s: // The coverage radius
$Boundary = \{f_1, \ldots f_a\}$;// The boundaries of the task area

Output: tmc;// The total moving cost

$tmc \leftarrow 0$; $T_{initcov} \leftarrow \emptyset$; $S_{initcov} \leftarrow \emptyset$;
for *each* t_i $(1 \le i \le m)$ **do**
  $S_{tmp} \leftarrow \emptyset$;
  for *each* s_j $(1 \le j \le n)$ **do**
    if $dist(t_i, s_j) \le r_s$ **then**
      $T_{inticov} = T_{inticov} \cup \{t_i\}$;
      $S_{tmp} = S_{tmp} \cup \{s_i\}$;
    $S_{inticov} = S_{inticov} \cup \{s_j | dist(t_i, s_j) \le dist(t_i, s_i), s_i \in S_{tmp}\}$
$T_{needcov} = T \setminus T_{initcov}$; $S_{rest} = S \setminus S_{inticov}$;
$Movecost$= **GEOCP**($T_{needcov}, S_{rest}, r_s, Boundary$);
$(P_{dest}, S_{move}, tmc)$ = **extended-Hungarian**($T_{needcov}$, S_{rest}, $Movecost$);
return tmc;

However, although the Basic algorithm minimizes the number of sensors to move, it may increase the total movement distance of sensors. For example, as shown in Fig. 3c, targets A and B could be covered by one single sensor. According to the Basic algorithm, s1 should be moved to cover them because it is closest to the intersection of the coverage disks of A and B among the three sensors. However, if we move two sensors s2 and s3 to cover tA and tB respectively, we can further reduce the total movement distance, although the number of moved sensors is not minimized. In the next section, we further propose a Voronoi diagram based algorithm to minimize the movement of sensors rather than the number of sensors to move.

The Target-Based Voronoi Greedy Algorithm: In this section, we present a target based Voronoi greedy algorithm (TV-Greedy) to minimize the total movement distance of sensors to cover targets.



(a) Initial position of targets and sensors.

(b) Voronoi graph of targets and groups of sensors.

(c) Moving decisions are made.

(d) Sharing a chief server.

Fig. 4. Illustration of the TV-Greedy algorithm.

Basic Idea and Definitions: The basic idea of TV-Greedy is to deploy the nearest sensor to cover the targets that are uncovered. Since sensors located in a target's Voronoi polygon are closer to this target than to others, we use Voronoi diagrams of targets to group sensors according to their proximity to the corresponding target. For the sake of clarity, the definitions and notations that will be used in the algorithm description is presented below:

- If a sensor is located in a target's Voronoi polygon, the sensor is defined as a server to this target, and the target is regarded as a client of its servers. The set of a target's servers is called that target's own server group (OSG). The sensor in a target's OSG that is nearest to the target is called the chief server of that target, and other sensors are called non-chief servers of the target.
- Two targets are neighbors if their Voronoi polygons share an edge. For two neighboring targets A and B, the sensor in A's OSG that is closest to B is called an aid server to B.
- A target's candidate server group (CSG) is the union of its own chief server and aid servers from neighbors. For a target, only sensors in its CSG will be dispatched to cover it.

Algorithm 2. The TV-Greedy Algorithm

Input: $T = t_1, t_2, \ldots, t_m$;//The position of all targe
$S = s_1, s_2, \ldots, s_n$;//The position of sensors
r_s;//The coverage radius
Output: tmc;//The total moving cost

Generate the Voronoi diagram (VD) of targets;
Determine neighbors for each target according to their Voronoi polygon;
Determine the OSG for each target according to S and VD;
for *each* OSG_i **do**
  Determine the $chief\ server$;
  Identify the aid server for t_i's neighbor;
for *each* t_i **do**
  if t_i *has already been covered* **then**
    Return $cost(t_i) = 0$;
  else
    Produce CSG_i of t_i;
    if $CSG_i \neq \emptyset$ **then**
      Move the nearest server to cover t_i;
      return $cost(t_i)$ = moving distance;
    else
      if *there exit neighbors' chief servers that could be shared* **then**
        move the nearest chief server to cover t_i;
        Return $cost(t_i)$= moving distance;
      else
        Regenerate the CSG of t_i by searching aid servers of the t_i's 2nd or higher order neighbors;
        Move the nearest aid server to cover t_i;
        Return $cost(t_i)$ = moving distance ;
  $tmc = tmc + cost(t_i)$
return tmc

For instance, as shown in Fig. 4, the own server group of target t_B is $OSG_B = \{s_4; s_5; s_6\}$, in which s_4 is the chief server. For other sensors in OSG_B, s_6 is the aid server for t_A, and s_5 is the aid server for t_C. Meanwhile, t_B has an aid server from t_C, which is s_2. Thus the candidate server group of t_B is $CSG_B = \{s_4; s_2\}$. Note that there is no sensor in target t_C's Voronoi polygon, and thus there is no aid server for t_B from t_C even though t_B and t_C are neighbors.

Algorithm Description: TV-Greedy starts from the generation of targets' Voronoi diagrams, which divides sensors into independent groups for each target. With assistance of targets' Voronoi diagrams, we can construct a sensor group for each target, which includes sensors in proximity to this target. Then, the nearest sensor to each target is selected from the target's group and its neighbors' group. After that, the selected sensor moves to the corresponding target. Details of the algorithm are described as follows, and its pseudo code is given in Algorithm 2.

First, the Voronoi diagram of targets is generated by using the coordinate information of targets which is known to sensors. Based on the vertices information of Voronoi polygons, the neighbors of each target are determined.

(Steps 1-2 in Algorithm 2): Second, the own server group OSG of each target is determined. In each OSG, the own servers (sensors in the OSG) is sorted by their distances to the client (the target of the OSG) in ascending order, according to which the chief server is identified as the first in the sorted list. For the rest own servers, we identify the aid server for each neighbor of the client via distance comparison and sorting, as shown in Fig. 4b (Steps 3-6 in Algorithm 2). Third, for each target, if it is covered initially, sensors in its OSG stand by and wait for orders (Steps 7-9 in Algorithm 2). If the target is not covered initially, then its CSG will be formed, which is a logical server group merged with the chief server of the target and all the aid servers from its neighbors. Then, if the CSG of a target is not empty, the nearest sensor is selected from the CSG to move (shown in Fig. 4c, Steps 10-15 in Algorithm 2). If the CSG is empty, it means that there is no sensor located in the target's Voronoi polygon. In this case, if there exit neighbors of the target that can share their chief server with the target, the nearest chief server moves to the nearest new position which is in the coverage disk of t_i' (shown in Fig. 4d, Steps 16-18 in Algorithm 2); otherwise, the CSG of t_i is regenerated by searching aid server of the 2nd order neighbor of t_i's (i.e., neighbors of neighbors) or higher order neighbor. After that, the nearest aid server moves to the nearest new position which is in t_i's coverage disk (Steps 19-22 in Algorithm 2).

V. SOLUTIONS TO THE NCON PROBLEM

The sensors that are used to cover targets in the TCOV problem are referred to as coverage sensors. After the TCOV problem is solved, all the targets are covered by at least one coverage sensor. Besides the coverage of targets in the first stage, another important requirement for a WSN is the connectivity of sensors and the sink, which promises the data transmission. If the sink and the coverage sensors are initially connected, then the connectivity problem is solved; otherwise, we need to study the NCON problem, i.e., how to connect the sink and the coverage sensors. The basic idea of providing connectivity is to relocate the rest mobile sensors to some locations where they can connect coverage sensors and the sink. Consider a tree-topology, where the sink is the root and all the coverage sensors are the leaf nodes, the goal of NCON is to relocate mobile sensors to new positions as intermediate node to connect the sink and coverage sensors, and the movement of sensor is minimized. From the above analysis, the NCON problem can be solved in two steps.

- First, we construct an edge length constrained Steiner tree spanning all the coverage sensors and the sink, such that each tree edge length is no longer than r_c. The Steiner tree is required to minimize the number of sensors that need to move.
- Second, we relocate the rest mobile sensors to the generated Steiner points to connect the coverage sensors and the sink. As for the second step, it is actually the special case of TCOV in which the Steiner points are regarded as "target"s and the coverage radius is zero. Then for each target we need to dispatch a dedicated sensor to cover it.

The key point to solve the NCON problem is to solve the first step: seeking an edge length constrained Steiner tree T spanning coverage sensors and the sink. Since the Steiner tree problem is NP-hard, we propose an approximate algorithm as follows: (1) constructing an euclidean minimum spanning tree (ECST), and (2) separating each edge of the spanning tree into the sections with length no longer than r_c. Because the sum of edge length in an euclidean minimum spanning tree is minimum, the number of section points on all edges is minimum. The euclidean minimum Spanning Tree algorithm is listed in Algorithm 3.

Algorithm 3. The ECST Algorithm

Input: $S = s_1, s_2, \ldots, s_n$;//The set of mobile sensors
S_{cov};//The set of coverage sensors
$sink(x, y)$;//The location of the sink
r_c;//The communication radius
Output: SP;//The set of Steiner points
$V = S_{cov}$;
Construct a complete graph $G = (V, E)$;
Construct an euclidean minimum spanning tree T_{ems} of G with the sink as the root;
for each $v_i \in V$ *and its parent* v_p^i **do**
Separate the edge $e(v_i, v_p^i)$ into $\lceil \frac{\|e(v_i, v_p^i)\|}{r_c} \rceil$ parts;
$SP(x_i, y_i) \leftarrow$ each separating point;
return SP;

With the output SP of the ECST algorithm, the next step is to assign the rest mobile sensors one-by-one to each point in SP with the minimum movement. Since it is actually an assignment problem, it can be solved using the extended Hungarian method described in Section 4.1. Algorithm 4 gives the ECST-Hungarian (ECST-H) algorithm to solve the NCON problem.

Algorithm 4. The ECST-H Algorithm

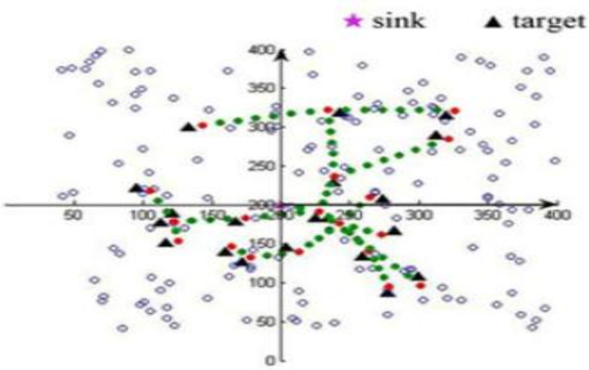
Input: $S = s_1, s_2, \ldots, s_n$;//The set of mobile sensors
S_{cov};//The set of coverage sensors
$sink(x, y)$;//The location of the sink
r_c;//The communication radius

1 ECST $(S, S_{cov}, sink(x, y), r_c)$;
2 extended-Hungarian $(SP,\ S/S_{cov})$; //Move the rest mobile sensors to the Steiner points
3 **return** movement cost and the deployment orders;

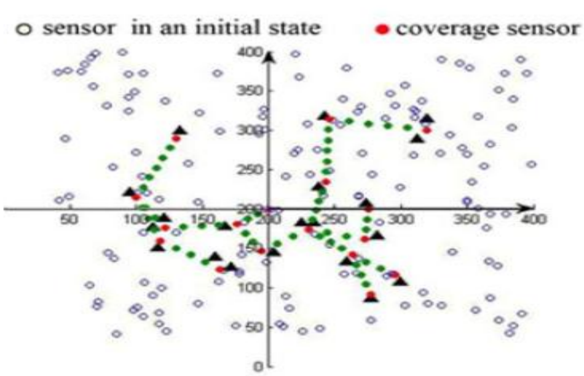
VII. SIMULATION EXPERIMENTS

A. Simulation Settings and Performance Metrics

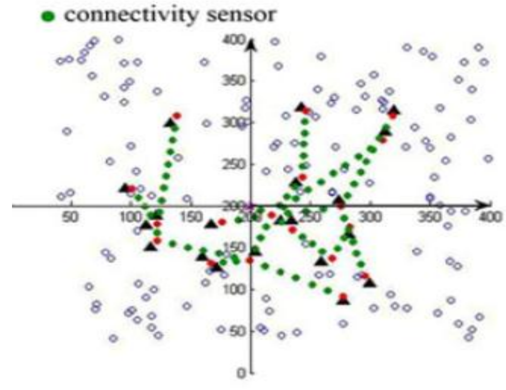
To evaluate the performance of the proposed algorithms, we conduct a set of simulation experiments by using Matlab. We first investigate the performance of the three solutions to the TCOV problem, namely Ex-Hungarian, Basic, and TV-Greedy, then study how their combinations with the ECST-H algorithm perform in solving the MSD problem. In the experiments, the targets and mobile sensors are randomly generated in a 400*400 m area. The default coverage radius and communication radius are r_s = 10 m and r_c = 15 m, respectively. For each combination of network parameters, we randomly generate 20 instances of the network and report the mean performance result. The primary metric concerned is the total movement distance of sensors. We consider two network parameters that



(a) Ex-Hungarian+ECST-H



(b) Basic+ECST-H



(c) TV-Greedy+ECST-H

Fig.5. Network topologies generated by different combinations of algorithms. There are 20 targets and 150 sensors in the network. Basic+ECST-Hmoves the least number of sensors, but TV-Greedy+ECST-H results in the shortest movement distance, as shown in Table 2.

may impact the movement distance of sensors: the number of targets (m) and the number of mobile sensors (n). As the TV-Greedy algorithm performs the best among the three algorithms, we also investigate the performance gap between TV-Greedy and the optimal solution in a small network to get an impression of how close TV-Greedy approaches the optimal solution.

B. An Illustration of Sensors Deployment

To get an intuitive impression of how our algorithms work, we demonstrate the generated tree topologies of the three different solutions, namely Ex-Hungarian+ECST-H, Basic +ECST-H, and TV-Greedy+ECST-H, in Fig. 5. There are 20 targets and 150 sensors in the network. We have the following observations. The first observation is that Basic uses the least number of coverage sensors and Ex-Hungarian uses exactly the same number of coverage sensors as targets because it moves sensors to targets in a one-to-one manner. The number of coverage sensors used by TV-Greedy is between the other two algorithms. Different choices of the coverage sensors also affect the choice of Steiner sensors in the NCON problem. As shown in Table 2, different solutions use different numbers of Steiner sensors to provide connectivity. In terms of the total number of both coverage sensors and Steiner sensors, Basic+ECST-H uses the least and TVGreedy+ ECST-H uses the most. The second observation is that TV-Greedy incurs much shorter movement distance than the other two solutions in the coverage stage. Although TV-Greedy uses more coverage sensors than Basic algorithm does (18 versus 16), its movement distance is much shorter (94.3 m versus 225.8 m). This owes to TV-Greedy's smart strategy in choosing coverage sensors. It groups sensors according to their proximity to the targets, and uses the nearest sensor to cover a target. This effectively reduces the movement distance to cover all the targets. It could also be observed that the differences in the solutions to the TCOV problem affect the performance of ECSTH in solving the NCON problem, which consequently impacts the overall performance of solutions to the MSD problem. ECST-H performs the best when TV-Greedy is used, and is much better than when the other two algorithms are used. Overall, the performance of TV-Greedy +ECST-H is the best among the three combinations.

C. Performance of Different Algorithms to TCOV

The Impact of the Number of Mobile Sensors: We first study how the number of mobile sensors affects the performance of the three solutions to the TCOV problem when the number of targets is fixed. Two scenarios are considered. In the first scenario, targets are scattered sparsely that the distance between any two targets is greater than $2*r_s$. In this case, the Ex-Hungarian method can find the optimal solution and thus can be used as the benchmark to evaluate the performance of Basic and TV-Greedy algorithms. In the second scenario, targets are scattered randomly and densely.

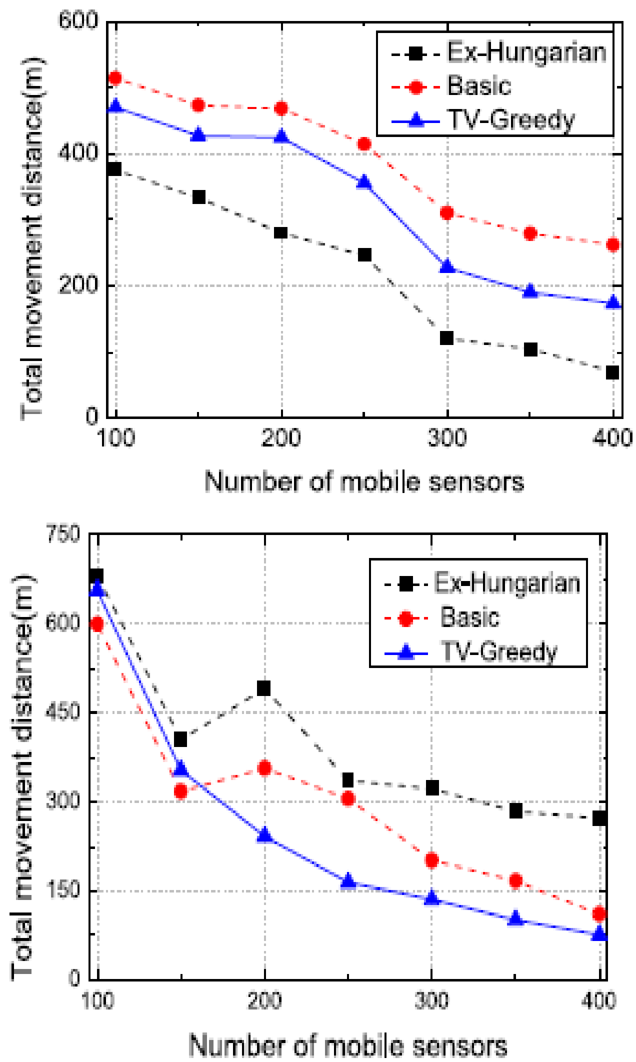


Fig. 6. Impact of the number of mobile sensors, n, on the movement distances of the three algorithms when m = 30: (a) When the distances between targets are greater than $2*r_s$; and (b) when targets are scattered randomly.

TABLE II: Performance Metrics of Different Combinations of Algorithms in Fig. 5

Algorithm Combination	Coverage/Steiner sensors	TCOV/NCON movement(m)	Total sensors/movement(m)
Ex-Hungarian+ECST-H	20/62	264.1/531.6	82/795.7
Basic+ECST-H	16/47	225.8/629.2	63/855
TV-Greedy+ECST-H	18/82	94.3/325.7	100/420

This represents the general case of TCOV in which the distances between targets might be less than $2*r_s$. Fig. 6a depicts the performance of the three algorithms in the first scenario when n varies from 100 to 400. We can see that when there are more sensors, all the three algorithms incur less movement distance for the following reason. With more sensors, each target can be covered by a closer sensor, which reduces the total movement distance. Ex- Hungarian performs best in this scenario, TV-Greedy follows, and Basic performs worst. Because the targets in this case disperse from each other more than double of communication radius, Ex-Hungarian can find the optimal solution. Compared to the optimal solution found by Ex- Hungarian, TV-Greedy and Basic incur about 25 percent more distance and 36 percent more distance, respectively. Obviously, TV-Greedy performs better than Basic does. The performance of the three algorithms in the general scenario is given in Fig. 6b. The trend is similar, i.e., all the three algorithms incur the shorter movement distance with a larger number of sensors. The difference from the first scenario is that TV-Greedy and Basic perform better than the Ex-Hungarian algorithm. This owes to the fact that in TV-Greedy and Basic algorithms different targets can be covered by the same sensor, while in Ex-Hungarian every target needs to be covered by distinct sensors. Thus TV-Greedy and Basic could use few sensors to achieve the coverage, which contributes to the reduction in movement distance. TV-Greedy performs nearly the same as Basic when the number of sensors is relatively small (e.g., $n \leq 100$), but outperforms Basic significantly when there are more sensors to dispatch (e.g., $n \geq 150$). Overall, TV Greedy is superior to both Ex-Hungarian and Basic in reducing movement sensors in the general case of TCOV.

The Impact of the Number of Targets: The impact of the number of targets on the movement distance of the proposed algorithms are also studied in the above-mentioned two scenarios. As shown in Fig. 7, the movement distance in both scenarios increases when m increases. The reason is that more targets need to be covered as m increases, which requires more sensors to be moved and consequently incurs longer movement distance. Fig. 7a shows the movement distance of different algorithms in the first scenario, i.e., when targets are spacing greater than 2*rs. In this scenario, Ex-Hungarian is optimal and performs best. TV-Greedy and Basic perform very close to each other and need about 35 percent more movement than EX-Hungarian. The results in the general scenario are shown in Fig. 7b. Again, TV-Greedy outperforms the other two algorithms, incurs up to 38 percent less movement distance than Basic and up to 21 percent less movement distance than Ex-Hungarian.

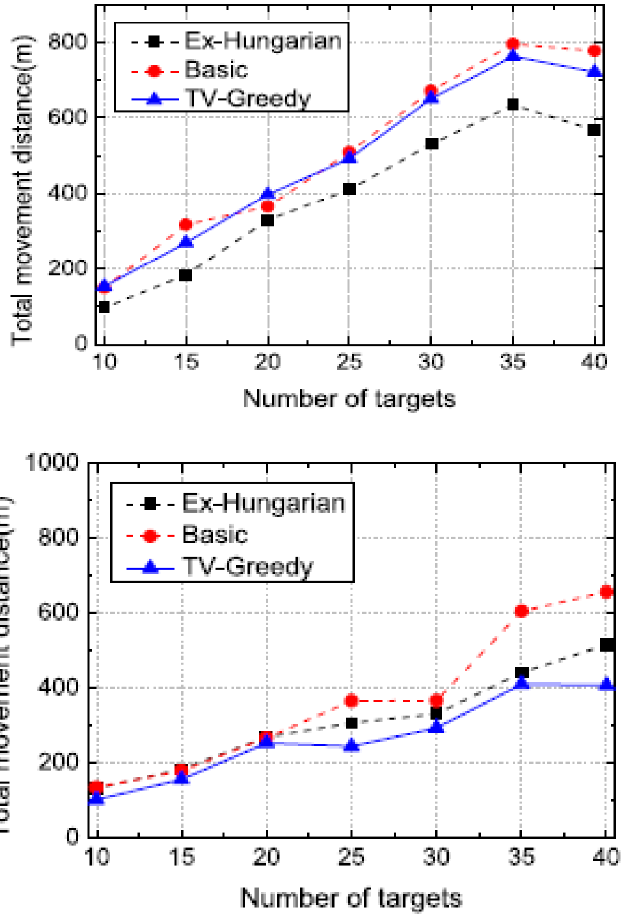


Fig.7. Impact of the number of targets,m, on the movement distances of the three algorithms when n = 300: (a) When targets are spacing greater than 2*r_s; and (b) when targets are scattered randomly.

VIII. CONCLUSION

In this work, we have studied the Mobile Sensor Deployment (MSD) problem in Mobile Sensor Networks (MSNs), aiming at deploying mobile sensors to provide target coverage and network connectivity with requirements of moving sensors. This problem is divided into two sub-problems, Target COVerage (TCOV) problem and Network CONnectivity (NCON) problem. For the TCOV problem, we prove it is NP-hard. For a special case of TCOV, an extended Hungarian method is provided to achieve an optimal solution; for general cases, two heuristic algorithms are proposed based on clique partition and Voronoi diagram, respectively. For the NCON problem, we first propose an edge constrained Steiner tree algorithm to find the destinations of mobile sensors, then use the extended Hungarian to dispatch rest sensors to connect the network. Theoretical analysis and simulation results have shown that, compared to extended Hungarian algorithm and Basic algorithm, the solutions based on TV-Greedy have low complexity and are very close to the optimum.

XI. REFERENCES

[1] R. Huang, W.-Z. Song, M. Xu, N. Peterson, B. Shirazi, and R. LaHusen, “Real-world sensor network for long-term volcano monitoring: Design and findings,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 2, pp. 321–329, Feb. 2012.

[2] G. Blumrosen, B. Hod, T. Anker, D. Dolev, and B. Rubinsky, “Enhancing rssi-based tracking accuracy in wireless sensor networks,” ACM Trans. Sens. Netw., vol. 9, no. 3, pp. 1–29, 2013.

[3] M. Lu, J. Wu, M. Cardei, and M. Li, “Energy-efficient connected coverage of discrete targets in wireless sensor networks,” in Proc. of 3rd Int. Conf. Netw. and Mobile Comput., pp. 43–52, 2005.

[4] B. Liu, O. Dousse, P. Nain, and D. Towsley, “Dynamic coverage of mobile sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 2, pp. 301–311, Feb. 2013.

[5] G. Wang, M. Z. A. Bhuiyan, J. Cao, and J. Wu, “Detecting movements of a target using face tracking in wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 4, pp. 939 949, 2014.

[6] R. Tan, G. Xing, J. Wang, and H. C. So, “Exploiting reactive mobility for collaborative target detection in wireless sensor networks,” IEEE Trans. Mobile Comput., vol. 9, no. 3, pp. 317– 332, Mar. 2010.

[7] Z. Fu and K. You, “Optimal mobile sensor scheduling for a guaranteed coverage ratio in hybrid wireless sensor networks,” Int. J. Distrib. Sens. Netw., vol. 2013, article no. 740841, pp 1–11, 2013.

[8] R. C. Luo and O. Chen, “Mobile sensor node deployment and asynchronous power management for wireless sensor networks,” IEEE Trans. Ind. Electron., vol. 59, no. 5, pp. 2377– 2385, May 2012.

[9] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, “Mobile element scheduling with dynamic deadlines,” IEEE Trans. Mobile Comput., vol. 6, no. 4, pp. 395–410, Apr. 2007.

[10] T. lin Chin, P. Ramanathan, K. K. Saluja, and K. Ching Wang, “Exposure for collaborative detection using mobile sensor networks,” in Proc. IEEE 2rd Int. Conf. Mobile Adhoc Sen. Syst., 2005, pp. 743–750.

[11] N. Bisnik, A. A. Abouzeid, and V. Isler, “Stochastic event capture using mobile sensors subject to a quality metric,” IEEE Trans. Robot., vol. 23, no. 4, pp. 676–692, Aug. 2007.

[12] L. He, J. Pan, and J. Xu, “A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements,” IEEE Trans. Mobile Comput., vol. 12, no. 7, pp. 1308–1320, Jul. 2013.

[13] X.-Y. Liu, K.-L. Wu, Y. Zhu, L. Kong, and M.-Y. Wu, “Mobility increases the surface coverage of distributed sensor networks,” Comput. Netw., vol. 57, no. 11, pp. 2348–2363, 2013.

[14] J. Luo, D. Wang, and Q. Zhang, “Double mobility: Coverage of the sea surface with mobile sensor networks,” ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 13, no. 1, pp. 52–55, 2009.

[15] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, “Cost-effective barrier coverage by mobile sensor networks,” in

Proc. IEEE 31st Annu. Int. Conf. Comput. Commun., 2012, pp. 819–827.

[16] G. Wang, M. J. Irwin, P. Berman, H. Fu, and T. L. Porta, “Optimizing sensor movement planning for energy efficiency,” in Proc. Int. Symp. Low Power Electron. Des., 2005, pp. 215–220.

[17] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. Sayrafian-Pour, “Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors,” in Proc. IEEE Int. Conf. Commun., 2011, pp. 1–6.

[18] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. S. Pour, “Distributed deployment algorithms for efficient coverage in a network of mobile sensors with nonidentical sensing capabilities,” IEEE Trans. Veh. Technol., DOI: 10.1109/TVT.2014.2302232, 2014.

[19] Y. Yang, M. I. Fonoage, and M. Cardei, “Improving network lifetime with mobile wireless sensor networks,” Comput. Commun., vol. 33, no. 4, pp. 409–419, 2010.

[20] Y.-C. Wang and Y.-C. Tseng, “Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage,” IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 9, pp. 1280– 1294, Sep. 2008.

[21] W. Luo, J. Wang, J. Guo, and J. Chen, “Parameterized complexity of max-lifetime target coverage in wireless sensor networks,” Theoretical Computer Science, vol. 518, pp. 32–41, 2014.

[22] J. Wang, W. Luo, Q. Feng, and J. Guo, “Parameterized complexity of min-power multicast problems in wireless ad hoc networks,” Theoretical Computer Science, vol. 508, pp. 16–25, 2013.

[23] E. Mathews and C. Mathew, “Deployment of mobile routers ensuring coverage and connectivity,” Int. J. Comput. Netw. Commun., vol. 4, no. 1, pp. 175–191, 2012.

[24] S. Zhou, M.-Y. Wu, and W. Shu, “Finding optimal placements for mobile sensors: wireless sensor network topology adjustment,” in Proc. IEEE 6th Circuits Syst. Symp. Emerging Technol.: Frontiers Mobile Wireless Commun., 2004, vol. 2, pp. 529–532.

[25] I. E. Korbi and S. Zeadally, “Energy-aware sensor node relocation in mobile sensor networks,” Ad Hoc Netw., vol. 16, no. 1, pp. 247– 265, 2014.

[26] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, “Deploying wireless sensors to achieve both coverage and connectivity,” in Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., 2006, pp. 131– 142.

[27] M. van de Vel, Theory of Convex Structures. Amsterdam, The Netherlands: North Holland, 1993, vol. 50.

[28] R. M. Karp, Reducibility among Combinatorial Problems. New York, NY, USA: Springer, 1972.

[29] E. L. Lawler, “The quadratic assignment problem,” Manage. Sci., vol. 9, no. 4, pp. 586–599, 1963.

[30] Z. Liao, S. Zhang, J. Cao, W. Wang, and J. Wang, “Min imizing movement for target coverage in mobile sensor networks,” in Proc. 32nd Int. Conf. Distrib. Comput. Syst., 2012, pp. 194–200.

[31] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Res. Logistics Quarterly, vol. 2, no. 1/2, pp. 83–97, 1955.

[32] R. Cole, “Parallel merge sort,” SIAM J. Comput., vol. 17, no. 4, pp. 770–785, 1988.

[33] Z. Liao, J. Wang, S. Zhang, and J. Cao, “Clique partition based relay placement in wimax mesh networks,” in Proc. IEEE Global Commun. Conf., 2012, pp. 2566–2571.

[34] M. de Berg, M. van Kreveld, M. Overmars, and O. C. Schwarzkopf, Computational Geometry. New York, NY, USA: Springer, 2000.

[35] M. I. Shamos, “Geometric complexity,” in Proc. 7th Annu. ACM Symp. Theory Comput., 1975, pp. 224–233.

[36] M. Dorigo and M. Birattari, “Ant colony optimization,” in Encyclopedia of Machine Learning, C. Sammut and G. I. Webb, Eds. New York, NY, USA: Springer, 2010, pp. 36–39.