Sweep Coverage with Mobile Sensors

Mo Li[1] Weifang Cheng[2] Kebin Liu[3] Yunhao Liu[1] Xiangyang Li[4] Xiangke Liao[2]

973 WSN Joint Lab

[1]Hong Kong University of Science and Technology, Hong Kong

[2]National University of Defense Technology, China

[3]Shanghai Jiao Tong University, China

[4]Illinois Institute of Technology, Chicago, USA

Email: wfangch@nudt.edu.cn limo@cse.ust.hk liu@cse.ust.hk xli@cs.iit.edu liao@nudt.edu.cn

Abstract

Many efforts have been made for addressing coverage problems in sensor networks. They fall into two categories, full coverage and barrier coverage, featured as static coverage. In this work, we study a new coverage scenario, sweep coverage, which differs with the previous static coverage. In sweep coverage, we only need to monitor certain points of interest (POIs) periodically so the coverage at each POI is time-variant, and thus we are able to utilize a small number of mobile sensors to achieve sweep coverage among a much larger number of POIs. We investigate the definitions and model for sweep coverage. Given a set of POIs and their sweep period requirements, we prove that determining the minimum number of required sensors (min-sensor sweep-coverage problem) is NP-hard, and it cannot be approximated within a factor of 2. We propose a centralized algorithm with constant approximation ratio 3 for the min-sensor sweep-coverage problem. We further characterize the non-locality of the problem and design a distributed sweep algorithm, DSWEEP, cooperating sensors to provide efficiency with the best effort. We conduct extensive simulations to study the performance of the proposed algorithms. Our simulations show that DSWEEP outperforms the randomized scheme in both effectiveness and efficiency.

Keywords: sweep coverage, mobile sensors, dynamic coverage, DSWEEP

Index Terms

sweep coverage, mobile sensors, dynamic coverage, DSWEEP

I. Introduction

Wireless sensor networks have been widely studied for environment surveillance applications. In such applications, achieving specific coverage requirements is essential. There has been tremendous work done for different coverage problems in sensor networks under two main existing coverage scenarios, full coverage and barrier coverage. In full coverage [1, 2, 3, 19, 24], sensors deployed over the field continuously monitor the entire area. Any point within the area is ensured to be covered by at least one or k sensors. A full coverage is required usually when users need to fully monitor the entire environment. In barrier coverage [4-6], sensors are deployed to form a barrier for detecting any intruders crossing the given strip area. Sensors cooperate to guard barrier coverage by covering the crossing paths. Barrier coverage is usually required for guarding safeties from intruders.

In either of the above two coverage scenarios, the monitored area requires being covered all the time, featured as static coverage. On the opposite, some applications set requirements with more dynamics along the time dimension. In a typical application of patrol inspection, we only need provide monitoring on certain Points Of Interest (POI) periodically instead of all along, which is featured as a ***sweep coverage***. Sweep coverage differs with the static coverage, in the sense that in sweep coverage the coverage at each POI is time-variant as long as a coverage period is guaranteed. Therefore, directly applying traditional work under static coverage to the sweep coverage scenario is not feasible, suffering from poor efficiency and unnecessary extra overhead.

In this work, we investigate the sweep coverage problem in sensor networks. We propose a model for sweep coverage, in which each POI is covered by a sensor at a specific time instance *iff.* the sensor is located at the position of the POI. A POI is t-sweep covered if it is covered at least once every t time units, and t is the sweep period of this POI. Different POIs could have different sweep periods. For periodical monitoring, we can utilize a small number of mobile sensor nodes to achieve sweep coverage among a much larger number of POIs. If stationary sensors are deployed, much more sensors are required and they need not work most of the time, leading to significant waste of sensor nodes. In this scenario, we assume that all sensors are mobile, since the situation consisting of both stationary and mobile sensors can easily be reduced to scheduling mobile nodes for sweep coverage among those POIs not covered by stationary sensors.

Given the sweep coverage model with a set of POIs and the requirement of their sweep periods, a natural problem is to determine the minimum number of mobile sensors for required sweep coverage, which we define as min-sensor sweep-coverage. Unfortunately, we prove that this min-sensor sweep-coverage problem is NP-hard and it cannot be approximated within a factor of 2 unless P = NP. It is even challenging whether we can design a polynomial algorithm achieving constant approximation ratio. We further characterize the non-locality of the sweep coverage problem, i.e., an individual mobile sensor cannot locally say "yes" or "no" to the question of whether a given set of POIs are globally t-sweep covered. As a result, how to design a sound distributed algorithm to cooperate the sensors achieving the sweep coverage efficiently is non-trivial.

We first target a simplified min-sensor sweep-coverage problem where the sweep periods of all POIs are assumed to be identical. We propose a centralized sweep algorithm, CSWEEP, to schedule the sensors, which has an approximation ratio $2 + \epsilon$ for any $\epsilon > 0$ on the minimum number of required sensors. Then we extend to general min-sensor sweep-coverage problem, and propose the GSWEEP algorithm, with an approximation ratio 3. In either CSWEEP or GSWEEP, the moving route of each mobile sensor is predetermined to guarantee the coverage. For practicability and scalability, we propose a distributed sweep algorithm, DSWEEP, which cooperates sensors efficiently to provide required coverage with the best effort. In DSWEEP, each sensor decides its moving path individually in runtime with the knowledge of the traces of others. Therefore, each sensor maintains a sweep table to save the swept POI ID and swept time. Sensors propagate their sweep tables to the network through the epidemic exchange. A filtered table exchange mechanism is utilized to omit transmitting most redundant table entries. Our simulations show that DSWEEP outperforms the randomized scheme in both effectiveness and efficiency.

The rest of this paper is organized as follows. Section II discusses related work. Section III describes the preliminaries on sweep coverage. We also prove the NP hardness of the min-sensor sweep coverage problem and present the centralized algorithms. In Section IV, we present the design of DSWEEP, including the information exchange and local decision processes. We conduct the performance evaluation of DSWEEP in Section V and finally, we conclude this work in Section VI.

II. Related Work

The coverage problem has been a hot issue in wireless sensor networks. Many efforts have been made on the full coverage problem, such as area coverage [10, 11] and point coverage [12]. There has been some work using mobile sensors to assist static coverage under a hybrid network architecture [13, 14]. Wang et al. investigate the optimized movement of mobile sensors to provide k-coverage in both mobile sensor networks and hybrid sensor networks [13]. The authors in literature [14] propose a distributed relocation algorithm, where each mobile sensor only requires local information to achieve optimal relocation. They explore the potentials of mobile sensors to extend the network lifetime. Also many researchers study the coverage of mobile sensor networks. Howard et al. [15] propose a potential-field-based algorithm and ensure that the initial configuration of nodes quickly spreads out to maximize coverage area. Wang et al. [16] present another virtual-force-based sensor movement strategy to enhance network coverage after an initial random placement of sensors. Sensor nodes are redeployed according to the virtual force calculation. They also consider the coverage holes in the network and move sensors to the desired target positions in order to improve the coverage [17]. Above algorithms aim to spread sensors over the field for a stationary configuration to maximize the coverage area. A complete survey of the full coverage problem is provided by Wu et al. [3].

Kumar et al. extensively study the barrier coverage problem [4-6], where the sensors form a barrier to prevent intruders from crossing a thin strip. The work in literature [4] is the first one to study the theoretical foundations of barrier coverage. A localized algorithm providing local barrier coverage is proposed in literature [5]. Balister et al. [6] further derive reliable density estimates for achieving barrier coverage and connectivity in thin strips.

Most of existing work focuses on static coverage with stationary configurations of the sensors. Even with mobile sensors, they mostly focus on achieving an optimized deployment through their mobility without exploring the dynamic coverage. Obviously, the results and approaches of the work do not directly apply to the sweep coverage scenario. One previous work [18] studies the dynamic aspects of the coverage in a mobile sensor network. It shows that while the area coverage at any given time instance remains unchanged, a larger area will be covered during a time interval. The targets that not detected in a stationary sensor network can now be detected by moving the sensors. However, it focuses on providing coverage for the full area and does not consider the sweep coverage scenario.

The concept of sweep coverage initially comes from the context of robotics [8, 9] which mainly concerns the metric of coverage frequency, i.e., the frequency of the coverage of each point. Robots coordinate or randomly move on the field and deploy communication beacons in the environment to mark previously visited areas. Robots then make local decisions on their motion strategy through communications with those beacons. The techniques proposed in the domain of robotics cannot be directly applied to sensor networks due to the highly integrated intelligence and costly hardware requirements of robots. To the best of our knowledge, this work is the first to introduce the sweep coverage in sensor networks which builds the theoretical foundation and proposes practical protocols.

III. The Sweep Coverage Problem

In this section, we first give some definitions of sweep coverage problem. We prove the NP hardness of determining the minimum number of sensors to provide required sweep coverage (min-sensor sweep-coverage problem). We find that this problem cannot be approximated within a factor of 2 unless P = NP. We then propose centralized approximation algorithms against the min-sensor sweep-coverage problem with constant approximation ratio. At the end of this section, we characterize the non-locality property of sweep coverage problem.

A. Sweep coverage

Assume that n mobile sensors

$$S = \{s_1, s_2, \cdots, s_n\}$$

are (randomly or strategically) utilized to monitor m points-of-interest (POIs)

$$H = \{h_1, h_2, \cdots, h_m\}$$

in a region.

Let $d_{i,j}$ be the Euclidean distance between POI h_i and h_j. We assume that all mobile sensors will move at the same speed v. At a specific time instance, a POI is covered by a sensor *iff.* the sensor is located at the position of that POI. We assume that all sensors are mobile, since the situation consisting of both stationary and mobile sensors can easily be reduced to scheduling mobile nodes for sweep coverage among those POIs not covered by stationary sensors.

Sweep coverage is different with traditional full coverage or barrier coverage in which users need provide static and continuous coverage all the time. In sweep coverage we only require that the POIs are covered at least once every certain time interval, so that we can guarantee event detection within a certain delay bound. Based on this, we define t-sweep coverage as follows.

Definition 1 (t-sweep coverage): A POI is said to be t-sweep covered by a coverage scheme $\mathcal{F}$ *iff.* it is covered at least once every t time units by the mobile sensors scheduled by $\mathcal{F}$.

Coverage scheme $\mathcal{F}$ is a schedule of the mobile sensor movement. If a POI is t-sweep covered, time interval t is called the *sweep period* of the POI. In practice, different POIs may have different sweep period requirements. We assume that the POI h_i need to be covered once every t_i time units.

Definition 2 (Global sweep coverage): A set of POIs are said to be globally sweep covered by a coverage scheme $\mathcal{F}$ *iff.* every POI h_i is t_i-sweep covered under $\mathcal{F}$.

When $t_i = t$ for all POIs, it becomes a simplified problem we call *global t-sweep coverage*.

B. Problem hardness

The most fundamental problem we concern is, given a set of POIs, ~~what is~~ the minimum number of mobile sensors ~~to~~ satisfy the required global sweep coverage under the t_i-sweep coverage constraints for each POI. We denote this problem as min-sensor sweep-coverage problem. We show by Theorem 1 that the min-sensor sweep-coverage problem is NP-hard by a reduction from the Traveling Salesman Problem (TSP).

Theorem 1: Given a set of POIs and their sweep coverage time-period requirement, ~~determining~~ the minimum number of required mobile sensors is NP-hard~~, and it cannot be approximated within a factor of 2 unless P = NP~~.

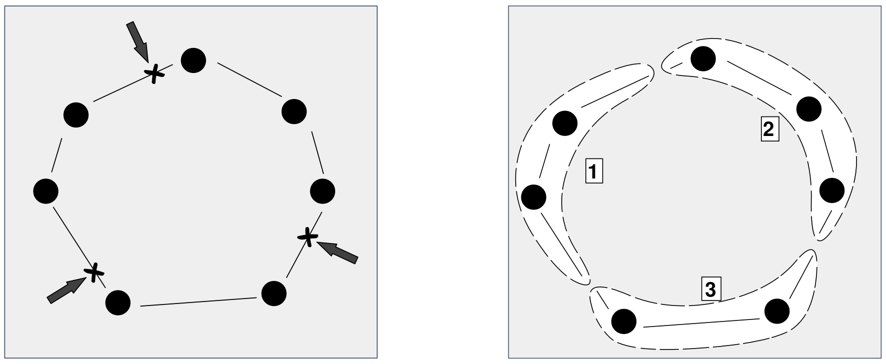
Proof: To prove the NP-hardness of the min-sensor sweep coverage problem, we reduce the TSP problem to the min-sensor sweep-coverage problem as follows.

~~Fo~~r a TSP problem, given a set of m sites $U = \{u_1, u_2, \cdots, u_m\}$ in a 2-dimensional domain~~,~~ TSP seeks the shortest route to visit all sites once and return to the starting point. The corresponding decision problem ~~of~~ TSP asks whether there is a cycle with length not exceeding a given value L. Given a decision problem of TSP (U, L), we define a min-sensor sweep-coverage problem accordingly: the POIs are ~~right~~ the m sites $U = \{u_1, u_2, \cdots, u_m\}$, and the sweep period t_i of each POI is $\frac{L}{v}$, where v is the moving speed of mobile sensors.

Apparently, if the given TSP problem $(U,\ L)$ has a solution, then one sensor is enough to provide $\frac{L}{v}$-sweep coverage[1]: the cycle that visits all sites defines a moving scheme $\mathcal{F}$ such that all sites will be visited by this sensor at least once every $\frac{L}{v}$ time units. On the other hand, if the min-sensor sweep-coverage problem has a solution of one sensor, the decision problem of TSP has a yes solution. ~~Because~~ for any interval of $t = \frac{L}{v}$ time units, each site must be visited at least once by this sensor during this time interval by the coverage scheme $\mathcal{F}$. This implies that the scheme $\mathcal{F}$ provides a route such that all sites are visited at least once. Obviously, the total length of this route is at most $\frac{L}{v} \cdot v = L$.

The above reduction proves that the min-sensor sweep-coverage problem is NP-hard. We then show that this problem does not have any polynomial time algorithm with approximation ratio $\leq 2 - \epsilon$ for an arbitrary $\epsilon > 0$, unless P = NP. For the sake of contradiction, assume that such a polynomial time approximation algorithm exists, denoted by APPR. Consider the decision TSP with L as the length of the optimum route for TSP. Then the corresponding min-sensor sweep-coverage still has optimum solution with one sensor. For this special min-sensor sweep-coverage

[1]In other words, the solution to this min-sensor sweep-coverage problem is 1.



(a) All the POIs are connected by the route computed by approximation algorithm PTAS of TSP. Then this route is divided into three equal pieces.

(b) Each mobile sensor is assigned to move continuously on one individual piece of route back and forth and monitors the POIs on its route.

Fig. 1. The illustration of CSWEEP algorithm.

problem, the number of sensors found by APPR will be at most $(2-\epsilon)\cdot 1$. It implies that the optimum solution for min-sensor coverage problem is 1, and this solution can be computed in polynomial time. This implies that the original TSP problem has a yes solution. Recall that, it is NP-hard to decide whether the decision TSP, with L as the length of the optimum route for TSP, has a yes solution. This finishes the proof. ■

C. *CSWEEP algorithm*

For the min-sensor sweep-coverage problem, global t-sweep coverage is a simplified case where $t_i = t$. For such case we design a centralized sweep algorithm (CSWEEP), which is derived from the approximation algorithm of the TSP problem.

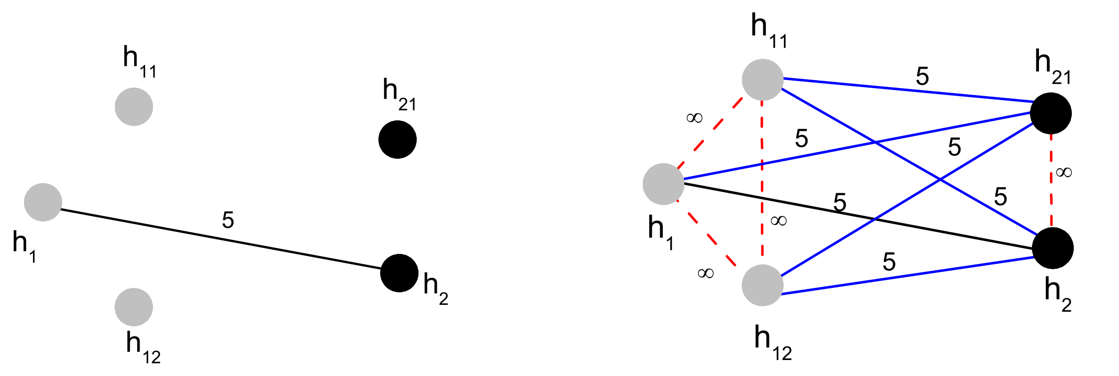
For the TSP problem, there is a well known polynomial time algorithm [21], PTAS, with the best approximation ratio $1+\epsilon$. We begin with this algorithm. First, we create a weighted complete graph using the given POIs as vertices, and the link weights is just the distance between two POIs.

We input this graph into PTAS. Then the output is a suboptimal route P for the corresponding TSP. Here every POI appears just once on P in the TSP problem. We partition route P into equal pieces with length $L_0 = \frac{v \cdot t}{2}$ as shown in Fig. 1(a). Then, we let each mobile sensor move continuously on one individual piece of route back and forth as shown in Fig. 1(b). As a result, each POI located on one piece of route will be visited at least once every $\frac{2 \cdot L_0}{v} = t$ time units. By this way, every POI is t-sweep covered and the set of POIs are globally t-sweep covered.

By Theorem 2 we further show that CSWEEP has an approximation ratio of $2 + \epsilon$.

Theorem 2: For the min-sensor sweep-coverage problem, the approximation ratio of CSWEEP algorithm is at most $2 + \epsilon$ for arbitrary $\epsilon > 0$.

Proof: First, taking the POIs of the min-sensor sweep-coverage as sites in TSP, we have the corresponding TSP problem. We assume that the length of optimal route for the TSP problem is L. ~~Notice that route P is derived from the algorithm PTAS~~. Then the length of route P is $L' = L \cdot (1 + \epsilon)$, since PTAS has an approximation ratio $(1 + \epsilon)$. Thus, the route P should be divided into [redacted] pieces in CSWEEP. As shown above, in CSWEEP we assign each mobile sensor an individual piece of route. Then the number of mobile sensors required in CSWEEP is $N_{cen} = \frac{2 \cdot L \cdot (1+\epsilon)}{v \cdot t}$. Second, we assume the optimal solution of min-sensor sweep-coverage problem is N_{opt}. In other words, there is a coverage scheme $\mathcal{F}$ and according to scheme $\mathcal{F}$, if we use N_{opt} sensors moving at constant speed v, each POI will be visited at least once in t time units. As L is the length of the shortest route for corresponding TSP problem, we get the following inequation $N_{opt} \cdot v \cdot t \geq L$ leading to $N_{opt} \geq \frac{L}{v \cdot t}$. Finally, the approximation ratio of CSWEEP is calculated $\frac{N_{cen}}{N_{opt}} \leq 2 + \epsilon$. This finishes the proof. ∎



(a) The link weight between h_1 and h_2 is right their distance 5. Two virtual POIs are derived for h_1, denoted by h_{11} and h_{12}. One virtual POI is derived from h_2, denoted by h_{21}.

(b) A complete graph among all POIs and virtual POIs is created. The link weights of the clique among $\{h_1, h_{11}, h_{12}\}$ are set to be ∞, and so does h_2 and h_{21}. Other link weights are duplicated from $d_{1,2}$.

Fig. 2. The illustration of duplicating POIs in GSWEEP.

D. GSWEEP algorithm

For the general case of min-sensor sweep-coverage problem, the sweep periods of different POIs might be different. Therefore, the above approximation cannot apply to such case and we design a general approximation algorithm, GSWEEP, executed in three steps.

Step 1. Duplicating the POIs. For each POI h_i, we calculate its monitoring frequency $f_i = \frac{1}{t_i}$. If f_i is not an integer, we convert it to integers by ceiling. Then we can compute the greatest common divisor of all the frequencies $f = \gcd(f_1, f_2, \cdots, f_m)$. For each POI h_i, we create $k(i) = \frac{f_i}{f} - 1$ virtual POIs for it, denoted by $H_i = \{h_{i1}, h_{i2}, \cdots, h_{ik(i)}\}$. As shown in Fig. 2(a), two virtual POIs, h_{11} and h_{12}, are derived for h_1. One virtual POI is derived from h_2, denoted by h_{21}. For all POIs and their virtual POIs, we create a weighted complete graph. First, the link weight between h_i and h_j is set the same as their distance $d_{i,j}$. All the link weights of the clique among h_i and POIs in H_i are set to be ∞. This implies that these links with ∞ weight do not exist in practice in the following algorithms. Whereas, the link weights for members of H_i between h_j and members of H_j are just duplicated from link weight between h_i and h_j. As shown in Fig. 2(b), the link weights of the clique among $\{h_1, h_{11}, h_{12}\}$ are set to be ∞, so does the link weight between h_2 and h_{21}. All the remaining link weights are set to be 5, duplicated from link weight between h_1 and h_2. In the following paper, we consider the virtual POIs the same as POIs.

Step 2. Finding a TSP route P. Since the above weighted graph is not a geometric graph, we cannot use the approximation algorithm PTAS to address the TSP problem on this graph, but with the help of Christofides algorithm [22], we can find a route P for this problem with an approximation ratio $\frac{3}{2}$, having a time complexity of $O(m^3)$ where m is the number of POIs. Notice that route P visits every POI just once and POI h_i has additional $k(i)$ duplicates on route P.

Step 3. Partitioning the route P. Similar with CSWEEP, we partition route P into some equal pieces, which have the length $L_0 = \frac{v}{2 \cdot f}$. Then we assign each piece of route one sensor moving on back and forth. In result, we can guarantee that all POIs including the virtual POIs on the route can be visited at least once in $\frac{1}{f}$ time units. Since POI h_i has additional $k(i)$ duplicates on route P, then h_i can be visited at least $k(i) + 1 = \frac{f_i}{f}$ times in $\frac{1}{f}$ time units. Therefore, during t_i time units, h_i is visited at least $\frac{f_i}{f} \cdot t_i \cdot f = 1$ times. Consequently, GSWEEP can guarantee the required sweep coverage.

Theorem 3: GSWEEP algorithm has an approximation ratio at most 3.

Proof: As shown in the GSWEEP algorithm, for the corresponding TSP problem on the complete graph we build, Christofides algorithm has an approximation ratio $\frac{3}{2}$. This implies that route P derived by Christofides algorithm has a length $L' \leq \frac{3 \cdot L}{2}$, if the length of optimal route of the TSP problem is L. Then the number of sensors required by GSWEEP is $N_{gs} = \frac{L'}{L_0} \leq \frac{3 \cdot L \cdot f}{v}$. At the same time, the optimal solution of the min-sensor sweep-coverage problem is $N_{opt} \geq \frac{L}{v \cdot \frac{1}{f}} = \frac{L \cdot f}{v}$. Therefore, the approximation ratio of GSWEEP is $\frac{N_{gs}}{N_{opt}} \leq 3$. This finishes the proof. ■

E. Non-locality of Sweep Coverage

In full coverage, it has been shown that sensors can locally determine whether a given region is not fully k-covered [2]. If any point on the perimeter of a sensor's sensing disk is covered by less than k sensors, then this sensor can locally conclude that the region is not fully k-covered.

In the case of sweep coverage, however, an individual mobile sensor cannot locally say "yes" or "no" to the question of whether a given set of POIs is globally sweep covered. We can explain this as follows.

In many applications, the number of POIs is large and the distance between them is long. One sensor is insufficient for many application requirements, and two or more mobile sensors are necessary. In such a mobile sensor network, if no centralized deterministic scheme like GSWEEP is provided, a sensor s_i cannot know the whole moving path of all other sensors. Then s_i cannot determine whether the POIs not monitored by itself during each sweep period have been visited by any other sensor during corresponding time period. Therefore, a sensor cannot locally determine whether all POIs are t-sweep covered. Consequently, t-sweep coverage cannot be guaranteed by any deterministic scheme $\mathcal{F}$ without global information. In other words, none of the distributed local algorithms can guarantee the required t-sweep coverage.

Unfortunately, centralized global algorithms are not scalable for large scale networks. In practice, the POIs to be sweep covered may change over time. Furthermore, the moving speed of mobile sensors might also vary and the mobile sensor may even fail during their trips. Therefore, both CSWEEP and GSWEEP are not scalable and adaptive to practical cases. To address these problems, we propose a distributed sweep algorithm, DSWEEP, using only local information to provide adaptive and reliable coverage with best effort of mobile sensors.

IV. THE DSWEEP ALGORITHM

As mentioned above, a distributed algorithm is necessary for manipulating large scale networks. Without centralized scheduled moving route, each sensor only locally decide its moving path on runtime based on the knowledge exchanged with other sensors. Two questions need be answered before launching the algorithm. How does one sensor exchange the information with other sensors in the dynamic network? And, how does one sensor decide which POI to move towards based on the obtained information? In this section, we describe the principle of DSWEEP in detail and answer above two questions.

A. Assumptions

DSWEEP makes following assumptions. All sensors know their instant locations on the 2-D plane, with the help of external location services such as GPS. Each POI has a globally unique position and ID. The positions and sweep period of all POIs are preknowledge for each sensor. Each sensor periodically sends out beacon messages, so each sensor knows the positions of all neighboring sensors. All sensors keep moving with constant speed. The communication range of each sensor is assumed to be larger enough so that the sensors can exchange their coverage information with neighboring nodes. Also, all sensors are assumed to be roughly synchronized [23].

B. Epidemic exchange

When a sensor arrives at one POI, it does the job of sampling and inspection. Then, it stores the coverage information, including the swept POI ID and swept time. All the POI ID and swept time pair forms a sweep table which is locally stored at the sensor. For the same POI, only the latest swept time is saved. In order to precisely determine the next POI, each sensor needs the global coverage information of all sensors. However, in a dynamic and mostly disconnected network, there are few connected paths for sensors to flood their sweep table.

To address this problem, we use a variant of epidemic routing [7] to exchange sweep tables among sensor nodes. Epidemic routing adopts a “store-carry-forward” paradigm: a node receiving a packet buffers and carries that packet as it moves, passing the packet on to new nodes that it encounters. Newly infected nodes, in turn, behave similarly. The random pairwise exchanges of messages among mobile hosts ensure eventual message delivery.

In our case, every time a mobile sensor encounters another one, they immediately exchange their sweep tables. And afterwards both of them locally combine the two sweep tables into a new table. The combining rules are as follows. If a new swept POI ID appears, the sensor just inserts it as a new entry in its own sweep table. If the same swept POI ID appears twice, the sensor only keeps the one with the latest swept time. Next time any two other sensors encounter, the same process is repeated, whereas exchanged tables are new ones. Therefore, the coverage information of a sensor can propagate quickly to the whole network. The ACK is used to guarantee reliable exchange process.

In fact, in above process, sensors do not need exchange the whole table with their neighbors. A sensor only needs those latest entries. For a sensor, however, it does not know what the neighbor has and what it needs before exchange. Therefore, we add a flag for each entry in the sweep

POI_ID	Swept_time	Sensor_ID
17	11,30	s_i
20	12,00	4
22	11,20	4
31	11,00	5
35	11,34	3
40	11,25	9

(a) The original sweep table of sensor s_j before exchange. Only one entry comes from sensor s_i.

POI_ID	Swept_time	Sensor_ID
17	11,40	1
20	11,40	s_j
31	11,10	8
35	11,50	8
40	11,25	s_j
52	11,25	1

(b) The original sweep table of s_i. Two entries come from s_j. s_i sends the shaded entries to s_j.

POI_ID	Swept_time	Sensor_ID
17	**11,40**	s_i
20	12,00	4
22	11,20	4
31	**11,10**	s_i
35	**11,50**	s_i
40	11,25	9
52	**11,25**	s_i

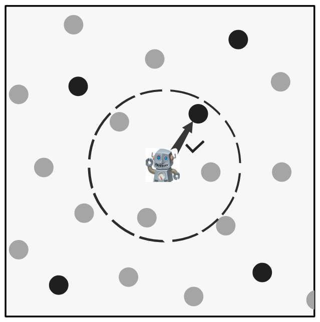
(c) Sweep table of s_j after combining entries received from s_i. s_j sends the shaded entries to s_i.

POI_ID	Swept_time	Sensor_ID
17	11,40	1
20	**12,00**	s_j
22	**11,20**	s_j
31	11,10	8
35	11,50	8
40	11,25	s_j
52	11,25	1

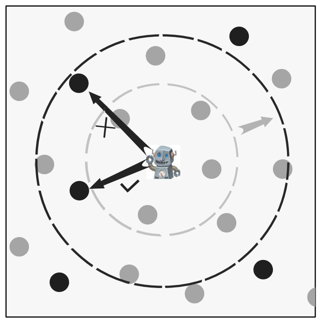
(d) Sweep table of s_i after combining entries received from s_j. The bold entries are new ones from s_j.

Fig. 3. An example of the filtered table exchange.

table, including the POI ID, swept time, sensor ID. The column sensor ID means the ID of the sensor where the latest swept time information of the POI comes from. Further, a sensor needs not send the neighboring node those entries from the neighbor itself. For example in Fig. 3, when s_i and s_j encounter, during setting up the connection, they exchange the number of entries in which the sensor ID is equal to its neighbor. Therefore, s_i knows the number of entries in which the sensor ID is equal to s_i in the table of s_j, denoted by n_1, and so does s_j, denoted by n_2. If n_2 is larger than n_1, sensor s_i first sends s_j the entries in which the sensor ID is not s_j, as shown in Fig. 3(a) and Fig. 3(b). After receiving the information from s_i, in Fig. 3(c), sensor s_j combines the entries into its own sweep table according to the above combination rules. Next it sends s_i the entries in which the sensor ID is not s_i, just like s_i did. Fig. 3(d) shows the new sweep table of s_i after epidemic exchange. Obviously, the later one to send the table entries can save quite a number of transmissions. We note that the sensor ID column will not be exchanged, since it is only used to indicate which the newest entry comes from. The filtered table exchange can filter most redundant entries between two neighbors. Therefore, the transmission overhead



(a) The sensor finds the next-POI with one-hop distance. Then the decision is done.



(b) The sensor finds two candidates with two-hop distance, and selects the more urgent one.

Fig. 4. An example of DSWEEP next-POI decision.

is largely reduced. For the example in Fig. 3, the number of exchanged entries is reduced from twelve to seven.

At the same time, the sensor periodically updates coverage information. Deleting outdated and useless information saves storage space and especially saves the energy consumption of data transmission. For each swept POI, if the time interval between its swept time and current time is no less than its sweep period, then it is outdated and deleted by the sensor.

C. *Next-POI decision*

After a sensor finishes sweeping one POI, it need decide the next POI to serve. The natural idea is that the nearest and most urgent POI should be first served. Considering the POIs in a planar graph, we can get the maximum distance between neighboring POIs, which is denoted as d_{max} and also referred to as one-hop distance. The moving speed is denoted as v. Therefore, the moving time of one-hop distance is $\frac{d_{max}}{v}$, which is also referred to as one-hop time. Similarly, $2 \cdot d_{max}$ is called as two-hop distance and $\frac{2 \cdot d_{max}}{v}$ is two-hop time.

When sensor s_j finishes sweeping POI h_i, it first checks the set of POIs less than one-hop distance from h_i, denoted as H_i. Then for each POI in H_i, sensor s_j checks its sweep time locally in the sweep table. If the ID of one POI is not in the sweep table, there are three cases. One is POI h_j has never been covered. The second is that POI h_i was swept long time ago, so its entry has been deleted by information updating. The third is sensor s_j has not obtained any coverage information of POI h_i. Both of the first two cases imply that POI h_j needs to be covered immediately. Therefore, the sensor marks these POIs as candidates. For all candidates, it chooses the closest one as next POI for saving energy. Otherwise, for each POI, its forthcoming sweep deadline is its last swept time added by its own sweep period. If the forthcoming deadline of any POI is within next one-hop time period, this POI is marked as an urgent POI. If multiple urgent POIs exist, the one with earliest sweep deadline is selected as next POI. If no POIs exist during the next one-hop time period, the sensor tries to find an urgent one during the next two-hop time period. Similarly, the sensor finds the POIs less than two-hop distance, and check whether their forthcoming sweep deadlines are within next two-hop time period. The same steps are repeated until its next POI is decided. The next-POI decision process is illustrated in Fig. 4. In Fig. 4(a), the sensor finds one candidate POI within the one-hop distance, and then it selects this POI as next-POI. In Fig. 4(b), the sensor finds no urgent POIs within one-hop distance, so it continues to check the stations within two-hop distance. Finally it finds two urgent candidate POIs in the forthcoming two-hop time. Then it selects the one with earliest deadline to move towards.

D. State transition of DSWEEP

To better describe the execution of DSWEEP, we analyze the state transition of DSWEEP in each sensor. As shown in the above, every sensor has five types of actions in DSWEEP.

- Exchange: the action of coverage information propagation described in section IV-B.

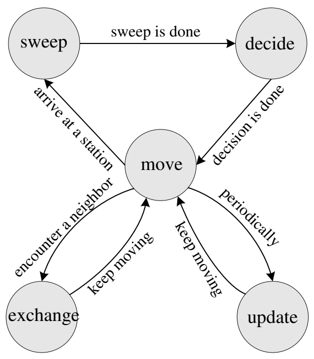


Fig. 5. State transition diagram of a mobile sensor.

- Update: the action of periodically checking the sweep table to delete outdated information described in section IV-B.
- Sweep: the action of patrol inspection at a POI.
- Decide: the action of determining the next POI to move towards, which is detailed in section IV-C.
- Move: the action of moving from one POI to another.

After deployment, all the sensors keep moving in the given region and perform the DSWEEP algorithm. The state transition of each sensor is shown in Fig. 5. In most of the time, the sensor keeps moving towards the targeted POI. When it arrives at the POI, it transits to the sweep state. The data sampling and inspection is performed, and then it starts to determine the next POI. After the next POI is determined, it moves towards it immediately. During moving in the network, if the sensor encounters another one, it will exchange its sweep table with the neighbor. At the same time, the sensor periodically updates its sweep table to delete dated information.

V. PERFORMANCE EVALUATION

We conduct simulation experiments on the 3d robot simulator simbad [20] to test the performance of our algorithms. We present the simulation results in this section.

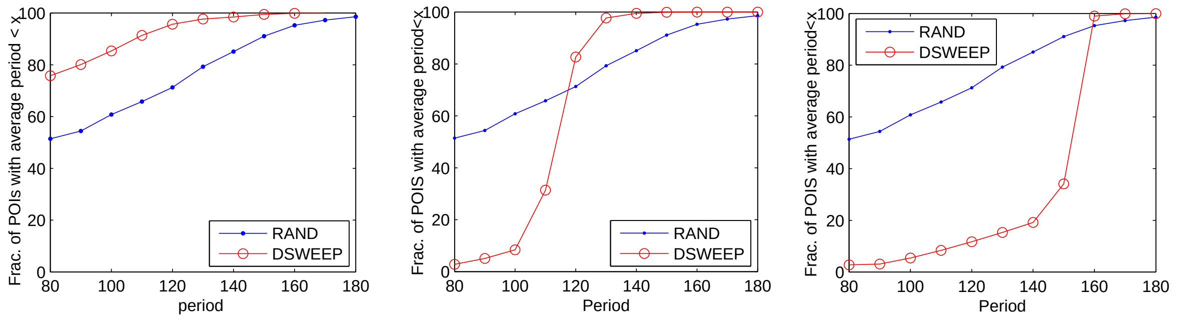
A. Simulation setup

For the simulations, we implement a sweep coverage instance on simbad [20]. 100 POIs are randomly deployed on a 10 meters by 10 meters square. The constant communication range of sensors is set to be 2 meters. The default moving velocity of mobile sensors is $0.3m/s$. Since the proposed sweep coverage is a purely new coverage scenario, existing distributed algorithms for sensor coverage could not directly apply to this scenario. Therefore, we propose a straightforward randomized scheme for comparison with our DSWEEP algorithm described in section IV. In the randomized scheme, each mobile sensor knows the positions of all POIs in advance. After the sensor arrives at a POI, it individually chooses a random neighboring POI as the next destination. For simplicity we name this randomized scheme as RAND in the following.

B. Coverage efficiency

We compare the coverage efficiency of DSWEEP and RAND under two different requirements of sweep coverage. One is all POIs require the same sweep period. The other is different POIs have different periods.

1) POIs with the same sweep period requirement: We set the same sweep period for all POIs in this subsection. The actual sweep period for each individual POI is the metric reflecting the coverage efficiency. Therefore, we first evaluate the cumulative distributed function (CDF) of the average sweep period for individual POIs. We also test the average sweep period of all POIs and the standard deviations.



(a) The situation when required sweep period $t = 80s$.

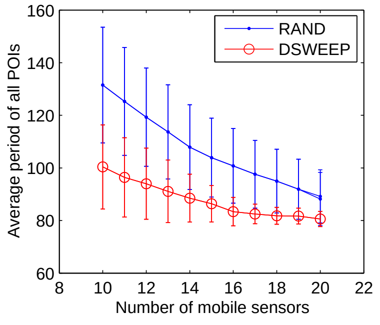
(b) The situation when required sweep period $t = 120s$.

(c) The situation when required sweep period $t = 160s$.

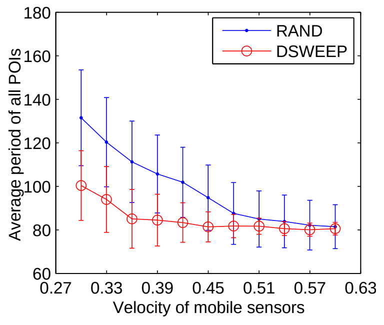
Fig. 6. The cumulative distribution function (CDF) of the average monitoring period of POIs ($n = 10$ and $v = 0.3m/s$).

We set the number of sensors $n = 10$ and the moving speed of mobile sensors to be $v = 0.3m/s$. Then for different required sweep periods $t = 80s$, $t = 120s$ and $t = 160s$, we do the following experiments respectively. We run the DSWEEP and RAND both for $100000s$ and compute the actual sweep period for each POI.

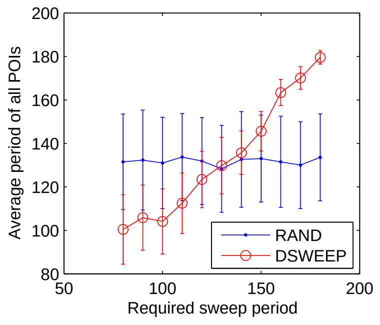
Fig. 6 shows how the sweep periods of the POIs vary with the required sweep period. Fig. 6(a) shows the CDF of different average periods of individual POIs when the required sweep period $t = 80s$. It is obvious that DSWEEP significantly outperforms RAND. First, for the fraction of POIs with average period less than $80s$, the required period, the result of DSWEEP is 78% much more than the 51% of RAND. This means, in DSWEEP more POIs meet their sweep period requirement. Furthermore, the CDF curve of DSWEEP reaches 100% more quickly than RAND which guarantees that for those POIs, which cannot meet their required sweep period, will not be delayed for too long. Fig. 6(b) presents the situation when the required sweep period $t = 120s$. Similarly with the previous situation, first we can find that the sweep periods of POIs in DSWEEP concentrate around the required sweep period, $t = 120s$, while those in RAND distribute along the entire span. Thus more POIs in DSWEEP fulfill the requirements and for



(a) Average period vs. the number of mobile sensors ($v = 0.3m/s$ and $t = 80s$).



(b) Average period vs. the velocity of mobile sensors ($n = 10$ and $t = 80s$).



(c) Average period vs. the required sweep period ($n = 10$ and $v = 0.3m/s$).

Fig. 7. The global average period of all POIs and standard variation by DSWEEP and RAND scheme.

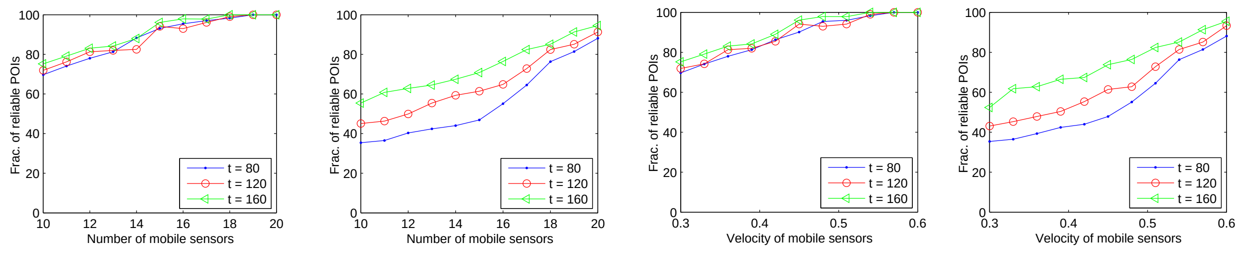
those exceeding the required period they will not be delayed for too long as in RAND. Fig. 6(c) lifts the required sweep period to be $160s$ and shows similar results. The main reason for above results is that the mobile sensor does not coordinate in the RAND scheme thus leading to the fact that some POIs might be visited frequently while other POIs might be visited rarely during a long time. In DSWEEP algorithm, however, if one POI h_i is monitored by a sensor recently, the sensor will try to send out the information through epidemic exchange. Thereafter, other sensors obtaining this information will not sweep cover it until the next deadline of POI h_i comes.

We further measure the average period of all POIs and the standard deviation. We compute the average period of all POIs to see the global effectiveness and calculate the standard deviation to see the fluctuation on individual POIs. We do three groups of experiments to evaluate the performance of RAND and DSWEEP in Fig. 7.

Fig. 7(a) varies the number of mobile sensors and plots the global average sweep period of all POIs. The moving speed $v = 0.3m/s$ and the required sweep period $t = 80s$. As expected, we see that both the global average period of DSWEEP and RAND decreases with the increase of the number of mobile sensors. The curve of DSWEEP is much lower than that of RAND and decreases quickly to $80s$, which means DSWEEP can guarantee most of the POIs meet their sweep period with much fewer sensors. The standard deviation of DSWEEP is always much smaller than that of RAND. A small standard deviation is very important to guarantee that the average sweep periods of most POIs are close to the global average difference and thus can fulfill the requirements. Fig. 7(b) varies the sensor velocity and plots the global average sweep period of all POIs. The number of mobile sensors $n = 10$ and the sweep period $t = 80s$. This result is similar with that in Fig. 7(a). Both the global average period of DSWEEP and RAND decreases with the increase of the velocity of mobile sensors. And as expected, DSWEEP outperforms RAND in terms of either small average sweep periods or small deviations. Fig. 7(c) varies the required sweep period. The number of mobile sensors $n = 10$ and the moving velocity $v = 0.3m/s$. As shown in the figure, apparently, the efficiency differs between RAND and DSWEEP. The average sweep period of RAND changes a little with the actual requirement while the average sweep period of DSWEEP is very sensitive to meet the varied requirement. Meanwhile, the standard deviations drop quickly which guarantees that the individual performance of most of the POIs are very close to the global capacity. Therefore, most of the POIs fulfill the required sweep period when the global capacity is adequate.

Through the above extensive simulations, compared with the randomized algorithm, DSWEEP provides required sweep coverage with fewer sensors under lower moving velocity.

2) POIs with different sweep periods: When the POIs have different importance, their required sweep periods can be different. In this group of experiments, we divide the POIs into three types: the first type with sweep period $t = 80s$, the second with $t = 120s$ and the third with $t = 160s$.



(a) Fraction of reliable POIs vs. the number of mobile sensors by DSWEEP ($v = 0.3m/s$).

(b) Fraction of reliable POIs vs. the number of mobile sensors by RAND ($v = 0.3m/s$).

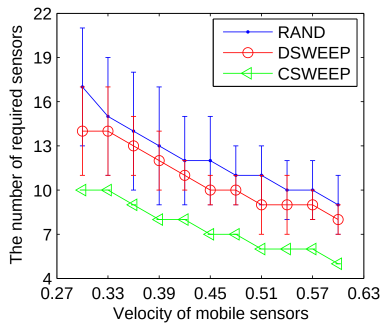
(c) Fraction of reliable POIs vs. the velocity of mobile sensors by DSWEEP ($n = 10$).

(d) Fraction of reliable POIs vs. the velocity of mobile sensors by RAND ($n = 10$).

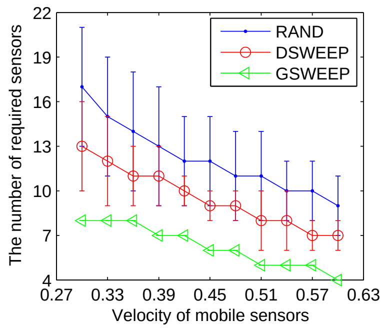
Fig. 8. The fraction of reliable POIs by DSWEEP and RAND scheme.

Each type has equal number of POIs. Then varied number of sensors and velocities are tested to evaluate their impact on the individual average period of POIs. We call the POIs which fulfill the required sweep period as reliable POIs. Fig. 8 shows the fraction of reliable POIs for three types of POIs respectively.

Fig. 8(a) and Fig. 8(b) compare DSWEEP and RAND with different number of mobile sensors. The moving velocity of mobile sensors is set to be $v = 0.3m/s$. Apparently DSWEEP outperforms RAND with a much larger number of reliable POIs. Moreover, in DSWEEP all three types of POIs have similar fraction of reliable POIs which shows the DSWEEP is adaptive to the hybrid sweep period requirements. In RAND, however, the three different types of POIs differ much with each other. The POIs with loose requirement ($t = 160s$) has a large fraction of reliable POIs but those with strict requirements ($t = 80s$) has only a small faction of reliable POIs. Similar results are shown in Fig. 8(c) and 8(d), where we vary the velocities of sensors. Therefore, according to above results, DSWEEP appears to be more adaptive and versatile to the hybrid sweep coverage requirements.



(a) The number of required sensors vs. the velocity with identical sweep period requirement $t = 80s$.



(b) The number of required sensors vs. the velocity with three types of sweep periods, $t = 80s, 120s, 160s$.

Fig. 9. The number of required sensors vs. various moving velocities by different algorithms.

C. The number of required sensors

We investigate the effectiveness on the min-sensor sweep-coverage problem in this section. The goal of the min-sensor sweep-coverage problem is to provide the required sweep coverage with the least number of mobile sensors. As mentioned above, no distributed local algorithms guarantee that every POI meets the sweep period requirement, neither does DSWEEP. Thus we test the actual average sweep period and compare it with the sweep period requirement. If with a relative error less than 10%, we consider the mobile sensors are eligible on providing the required sweep coverage. Fig. 9(a) shows the required number of mobile sensors by RAND, DSWEEP and CSWEEP under the identical sweep period requirement for all POIs $t = 80s$. Fig. 9(b) shows the required number of mobile sensors under three different sweep period requirements for the POIs, i.e., $t = 80s$, $120s$ and $160s$, by RAND, DSWEEP and GSWEEP. As the velocity of mobile sensors increases, all algorithms need fewer sensors. The CSWEEP and GSWEEP as global centralized algorithms set the lower bounds for DSWEEP, whereas, DSWEEP always outperforms RAND.

All the above experiments show that the proposed distributed algorithm DSWEEP outperforms the randomized scheme in both effectiveness and efficiency, whereas the proposed centralized algorithms outperforms DSWEEP in the number of required sensors.

VI. Conclusion

Patrol inspection with mobile sensors is an efficient scheme for many environments surveillance applications with specified delay bounds. We define the concept of sweep coverage to model the requirements of periodically monitoring a set of POIs in such applications. We discuss the problem of determining the minimum number of required sensors for given sweep coverage requirements. We prove that this min-sensor sweep-coverage problem is NP-hard and it cannot be approximated within a factor of 2. Accordingly we propose a general centralized algorithm, GSWEEP, with constant approximation ratio 3 for this problem. We further design a distributed sweep algorithm, DSWEEP, which cooperates sensors to provide efficient sweep coverage for given POIs and their sweep period requirements with the best effort. The simulation results show that DSWEEP outperforms a straightforward randomized scheme in both effectiveness and efficiency.

Sweep coverage is a purely new concept for sensor network monitoring. There are still many interesting problems not discussed in this paper. One significant extension of this problem is that for a given area rather than a set of discrete POIs, how to determine the metric of sweep coverage and study the applicability? How to work towards a bounded distributed algorithm and reduce the communication cost in a practical protocol for sweep coverage is also challenging. In our future work, we plan to study these problems and obtain more useful results.

REFERENCES

[1] S. Kumar, T. H. Lai, and J. Balogh, “On k-Coverage in a Mostly Sleeping Sensor Network,” in Proceedings of ACM MobiCom, 2004.

[2] C. F. Huang and Y. C. Tseng, “The Coverage Problem in a Wireless Sensor Network,” in Proceedings of ACM WSNA, 2003.

[3] M. Cardei and J. Wu, “Energy-efficient Coverage Problems in Wireless Ad Hoc Sensor Networks,” Journal of Computer Communications on Sensor Networks, 2005.

[4] S. Kumar, T. H. Lai, and A. Arora, “Barrier Coverage With Wireless Sensors,” in Proceedings of ACM MobiCom, 2005.

[5] A. Chen, S. Kumar, and T. H. Lai, “Designing Localized Algorithms for Barrier Coverage,” in Proceedings of ACM MobiCom, 2007.

[6] P. Balister, B. Bollobas, A. Sarkar, and S. Kumar, “Reliable Density Estimates for Achieving Coverage and Connectivity in Thin Strips of Finite Length,” in Proceedings of ACM MobiCom, 2007.

[7] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Technical Report CS-200006, Duke University 2000.

[8] D. W. Gage, “Command Control for Many-robot Systems,” in Proceedings of AUVS Technical Symposium, 1992.

[9] M. A. Batalin and G. S. Sukhatme, “Multi-robot Dynamic Coverage of a Planar Bounded Environment”,in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.

[10] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, “Wireless Sensor Networks with Energy Efficient Organization,” Journal of Interconnection Networks, 2002.

[11] S. Slijepcevic and M. Potkonjak, “Power Efficient Organization of Wireless Sensor Networks,” in Proceedings of IEEE ICC, 2001.

[12] M. Cardei and D.-Z. Du, “Improving Wireless Sensor Network Lifetime through Power aware Organization,” ACM Wireless Networks, vol. 11(3), 2005.

[13] W. Wang, V. Srinivasan, and K. C. Chua, “Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks,” in Proceedings of ACM MobiCom, 2007.

[14] D. Wang, J. Liu, and Q. Zhang, “Field Coverage using a Hybrid Network of Static and Mobile Sensors,” in Proceedings of IEEE IWQoS, 2007.

[15] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem,” in Proceedings of DARS, 2002.

[16] G. Wang, G. Cao, and T. L. Porta, “Sensor Deployment and Target Localization based on Virtual Forces,” in Proceedings of IEEE Infocom, 2003.

[17] G. Wang, G. Cao, and T. L. Porta, “Movement-assisted Sensor Deployment,” in Proceedings of IEEE Infocom, 2004.

[18] B. Liu, P. Brass, and O. Dousse, “Mobility Improves Coverage of Sensor Networks,” in Proceedings of ACM MobiHoc, 2005.

[19] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. “Coverage Problems in Wireless Ad-hoc Sensor Networks,” in Proceedings of IEEE Infocom, 2001.

[20] Simbad, http://simbad.sourceforge.net/.

[21] S. Arora, “Polynomial-time Approximation Schemes for Euclidean TSP and Other Geometric Problems,” in Proceedings of IEEE FOCS, 1996.

[22] Christofides, N, “Worst-case Analysis of a New Heuristic for the Travelling Salesman Problem,” Technical Report 388, Carnegie Mellon University, 1976.

[23] B. Sundararaman, U. Buy and A.D. Kshemkalyani, “Clock Synchronization in Wireless Sensor Networks: A Survey,” Ad-Hoc Networks, 3(3), May 2005.

[24] L. Lin and H. Lee. “Distributed Algorithms for Dynamic Coverage in Sensor Networks,” in Proceedings of ACM PODC, 2007.