On Timely Sweep Coverage with Multiple Mobile Nodes

Di Zhang, Dong Zhao, and Huadong Ma
Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing University of Posts and Telecommunications, Beijing 100876, China
iedizhang@163.com, {dzhao, mhd}@bupt.edu.cn

***Abstract*—Sweep coverage uses mobile nodes for monitoring Points of Interests (PoIs) in a sensing field. With sweep coverage, information can be effectively gathered without using a lot of static sensors. In this paper, we study the sweep coverage problem with sensing and transmission delay constraints, which is regarded as the Timely Sweep Coverage problem. Specifically, we investigate how to use the minimum number of mobile nodes to cover all PoIs under the two delay constraints. We propose two heuristic algorithms, MR-MinExpand and CoTSweep, to provide timely sweep coverage under different scenarios. MR-MinExpand leverages the difference between sensing and transmission delay constraints to schedule the route for each mobile node. CoTSweep considers the scenario where some PoIs cannot be covered by a single mobile node, and leverages the collaboration between mobile nodes to enable the sink node to collect data from remote PoIs. Extensive simulations and comparisons with previous works are conducted to validate the advantages of our algorithms.**

I. Introduction

In the past, stationary *Wireless Sensor Networks* (*WSNs*) have been extensively investigated for various applications such as environmental monitoring, habitat monitoring and forest fire response. Generally, it is necessary to deploy a large number of redundant sensors to maintain both coverage and network connectivity, which is often infeasible or undesirable due to a large cost, especially in a large-scale sensing region.

On the other hand, node mobility can be exploited to greatly improve the capability of sensing coverage and data transmission, resulting in novel data collection paradigms, such as delay-tolerant/opportunistic sensor networks and mobile crowd-sensing networks [1]–[3]. However, the random mobility of nodes carried by animals, humans or vehicles always results in uncertain data collection delays. In this paper, we focus on another type of mobile sensing applications such as patrol inspection and message ferrying, which require to provide monitoring or data collection services for a specified set of *Points of Interest* (*PoIs*) periodically under a certain delay constraint. Such applications actively control mobile nodes to move around to collect data from PoIs, and often achieve various objectives, such as minimizing the movement speed [4] or the number of mobile nodes [5]–[8] under a delay constraint, and finding the shortest period or trajectory length [9]–[13] given the number of PoIs. Such problems are often characterized as *Sweep Coverage*.

At present, most research on sweep coverage [5]–[7], [9]–[14] considers only the *sensing delay constraint* (*SDC*), which characterizes how often a PoI can be covered or visited by one mobile node, but fails to account for the *transmission delay constraint* (*TDC*), which characterizes how soon the collected data can be delivered to the sink node. In fact, the transmission delay is also significant in many scenarios, because the collected data must be offloaded timely due to the limited storage of mobile nodes and the guarantee of data freshness. For instance, in a marine monitoring application illustrated in Fig. 1, *Unmanned Aerial Vehicles* (*UAVs*) are used to collect data periodically from sensing buoys sparsely deployed in a large-scale marine environment, and deliver data timely to a ground base station. Each UAV has a limited storage, and different data types have different requirements of timeliness - e.g., the water quality data may tolerate a higher transmission delay just for analysis afterwards, whereas the event detection data should be delivered to the base station as soon as possible. Moreover, the transmission delay will affect the sweep coverage inevitably, as mobile nodes must take extra time to visit the sink.

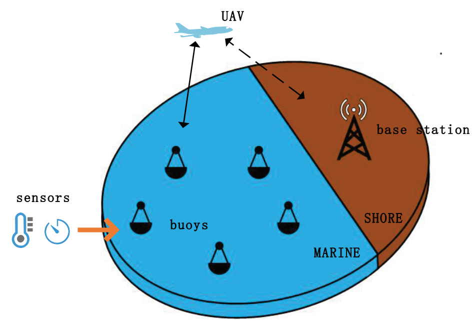


Fig. 1: Illustration of a marine monitoring application.

Our earlier work [4] is the first to present the concept of *timely sweep coverage* by integrating both SDC and TDC, but it considers only one single mobile node. In this paper, we consider multiple nodes. Specifically, we focus on the *Min-Nodes Timely Sweep Coverage* (*MNTSC*) problem, aiming to minimize the number of mobile nodes required to guarantee that each PoI is covered at least once by a mobile node within its SDC, and the collected data is delivered to the sink node within its TDC.

In this paper, we first prove that the MNTSC problem is NP-hard and then we propose two heuristic algorithms, *Multiple Return MinExpand Algorithm* (*MR-MinExpand*) and *Cooperative Timely Sweep Coverage* (*CoTSweep*), to solve the MNTSC problem. Considering the two delay constraints for

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

each PoI, a natural method to solve this problem is to use the smaller one of the two delay constraints to allocate PoIs for mobile nodes. However, this does not make effective use of the two delay constraints. In our proposed MR-MinExpand algorithm, according to the difference between SDC and TDC, we adopt the strategy of making the movement path of the mobile node pass through the sink node more than once. When the movement path passes through the sink node, the previously collected data can be submitted, so the TDCs of PoIs on the previous movement path are satisfied and can be ignored when generating the remaining movement path. In the special scenario where some PoIs are far from the sink node, so that the constraints cannot be satisfied, we design the CoTSweep algorithm to leverage the collaboration between mobile nodes to cover those PoIs. We also evaluate our proposed algorithms with two existing sweep coverage algorithms to validate the advantages of our algorithms.

The reminder of this paper is organized as follows. Section 2 reviews the related work. Section 3 provides the problem formulation. Section 4 and 5 present the MR-MinExpand and CoTSweep algorithms respectively. In Section 6, we evaluate our algorithms. Finally, we conclude this paper in Section 7.

II. Related work

The sweep coverage problem has attracted much attention over the past years. The authors of [6], [15] studied the problem of finding the minimum number of mobile nodes in sweep coverage. They converted the problem into the TSP problem and proposed two centralized algorithms. The authors of [7] also investigated the problem of finding the minimum number of mobile nodes in sweep coverage and proposed the MinExpand algorithm, which gradually deployed PoIs to the mobile node according to the path increment. In addition, they also proposed an algorithm called the OSWEEP algorithm, which was an improvement of the CSWEEP algorithm [6], [15]. The authors of [11] proposed algorithms to find the shortest trajectory length for saving the energy of mobile nodes. In their algorithm, they took the communication range of PoIs into consideration. The authors of [12] investigated the problem of minimizing the longest sweep period. In addition, they also proposed an algorithm named PathSplit to solve a variant problem of sweep coverage where the target was to cover all the given edges.

In this paper, we focus on the problem of using the minimum number of mobile nodes in timely sweep coverage. Note that most of the previous studies did not consider the timeliness of the collected data. That is, in their proposed schemes, they did not consider the data submission of mobile nodes. In fact, the TDC limits the longest distance that a mobile node can move. A small amount of existing work considers mobile node returning to the sink node. In [4], the authors considered the transmission time of the collected data, and investigated the minimum speed problem. The authors of [16] considered the energy constraint when studied the minimum mobile nodes problem. The energy constraint makes the mobile node need to return to the base station periodically. The authors of [8] also proposed algorithms to take the return time of mobile nodes into account. However, these algorithms focus on the scenarios without remote PoIs. In this paper, we propose an algorithm which can cover the remote PoIs. Besides, in our algorithms, we leverage the difference between SDC and TDC to make our algorithms more efficient.

III. Problem formulation

Assume that, in a sensing field, there are n PoIs $V = \{p_1, p_2, ..., p_n\}$ and a sink node u whose coordinates have been known. In our model, a PoI is covered when a mobile node moves to the position of the PoI. Mobile nodes move at the same speed v. The distance between PoI p_i and PoI p_j is defined as the Euclidean distance between their coordinates, denoted by $dist(p_i, p_j)$. $len(p_i, p_j)$ indicates the distance between PoI p_i and PoI p_j on the movement path. For arbitrary PoI p_i, $D(T_{s_i}, T_{tr_i})$ denotes its delay constraint, where T_{s_i} and T_{tr_i} denote the SDC and TDC, respectively. It is easy to get that T_{tr_i} needs to meet $T_{tr_i} \geq \frac{dist(p_i,u)}{v}$, otherwise, PoI p_i's TDC cannot be satisfied.

As defined in [4], a PoI p_i is said to be timely sweep covered, if and only if its SDC T_{s_i} and TDC T_{tr_i} can be satisfied by at least one mobile node. In the MNTSC problem, a coverage scheme is to make sure that each PoI $p_i \in V$ is timely sweep covered by at least one mobile node if possible, which is regarded as the *Global timely sweep coverage* (*GTSC*).

In our proposed algorithm, we will design a timely sweep coverage scheme F that leverages the minimum number of mobile nodes to cover the PoIs $p_i \in V$. Formally, the problem of finding the minimum number of mobile nodes in timely sweep coverage is defined as follows.

Definition 1 (Min-Nodes Timely Sweep Coverage, MNTSC)**.** *Given a set of PoIs $V = \{p_1, p_2, ..., p_n\}$ and each PoI p_i is associated with an SDC T_{s_i} and a TDC T_{tr_i}. The goal of the MNTSC problem is to find a scheduling scheme F, which requires the minimum number of mobile nodes, while satisfying the global timely sweep coverage.*

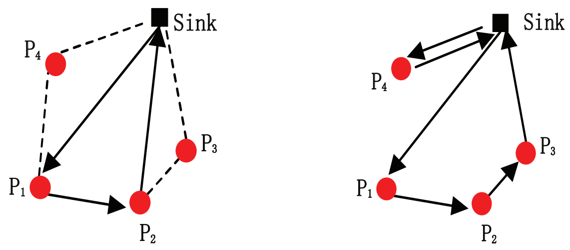
The MNTSC problem can be proved to be NP-hard. Theoretically, different PoIs may have different SDC and TDC. In this paper, we focus on a simplified version of the MNTSC problem and assume that all PoIs have the same delay constraints $D(T_s, T_{tr})$, i.e.,

$$\begin{cases} T_{s_i} = T_{s_j} = T_s, \\ T_{tr_i} = T_{tr_j} = T_{tr}, \end{cases} \quad 1 \leq \forall i, j \leq n. \tag{1}$$

Theorem 1. *The MNTSC problem is NP-hard.*

Proof. In the MNTSC problem, we consider the special case where all PoIs have the same delay constraint $D(T_s, T_{tr})$. According to the definitions of TDC and SDC, the TDC determines the longest distance between PoIs and the sink node on the movement path. Besides, the movement path for sensing is longer than the movement path for transmission.

Suppose the distance between the sink node and all PoIs in the sensing field is the same, which is denoted by $dist$.



(a) (b)

Fig. 2: Illustration of MR-MinExpand algorithm.

In this case, all PoIs are considered as being distributed on a circle with a radius of $dist$, and the sink node is the center. In this scenario, we set $T_s = T_{tr} + \frac{dist}{v}$. It is easy to prove that the MNTSC problem can be converted to the ordinary sweep coverage problem, which only has one delay constraint. According to [15], the problem of finding the minimum number of mobile nodes required in ordinary sweep coverage is NP-hard. Note that it is a special version of the MNTSC problem. Therefore, this proves that the MNTSC problem is NP-hard. □

IV. MULTIPLE RETURN MINEXPAND ALGORITHM (MR-MINEXPAND)

In the MR-MinExpand algorithm, mobile nodes are added in the sensing field through iterative processes. In each iteration, the mobile node tries to expand its movement path to visit more uncovered PoIs until the delay constraints cannot be satisfied. The key points of the algorithm are how many mobile nodes are needed to cover all PoIs and how to schedule the mobile nodes' movement paths.

The details of the MR-MinExpand algorithm are described in Algorithm 1. The basic idea of the MR-MinExpand algorithm is to select a PoI from the uncovered PoI set V' with the smallest path increment under the SDC. Then, verifying whether the TDC is satisfied. If both SDC and TDC can be satisfied, then this PoI is added into the path. Otherwise, we assign a new mobile node in the sensing field. Suppose the length of the current mobile node's movement path is denoted by $|L_m|$, the MR-MinExpand algorithm chooses the PoI with the maximum remaining delay constrain each time. That is, the chosen PoI satisfies

$$max(T_s - \frac{|L_m| + \Delta_{p_i}}{v}), \tag{2}$$

where Δ_{p_i} is the path increment after inserting PoI p_i. As shown in Fig. 2(a), the path increment of PoI p_3 is $\Delta_{p_3} = len(p_2, p_3) + len(p_3, u) - len(p_2, u)$.

However, only using the strategy mentioned above is not an efficient way to use the mobile node in timely sweep coverage. In fact, when $T_s \geq T_{tr}$, if the TDC is not satisfied, PoIs can still be added to the current mobile node's movement path. As shown in Algorithm 1, after selecting a candidate PoI p_i, the MR-MinExpand algorithm will check whether the TDCs of all the PoIs on the current mobile node's movement path are satisfied. If some PoIs' TDCs are not satisfied, the algorithm

Algorithm 1 Multiple Return MinExpand Algorithm, MR-MinExpand

Input: The PoI set V, the sink node u, the SDC T_s, and the TDC T_{tr}.
Output: The number of mobile nodes required m and the sweep paths L_1,L_2,...,L_m.

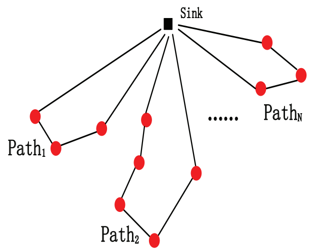
Set $m \leftarrow 1$, the already covered PoI set $V_c = \emptyset$. Find a PoI with the longest distance from u and put it into L_1.
while $V_c \neq V$ **do**
  for $edge \in L_m$ **do**
    Under the SDC T_s, find the candidate PoI p_k with the maximum remaining constraint.
    if the TDC T_{tr} is satisfied **then**
      Suppose PoI p_i and PoI p_j are the two end points of the edge. Insert PoI p_k between p_i and p_j.
      Put PoI p_k into V_c
    else if $T_s \geq T_{tr}$ **then**
      Make L_m pass through u and select the uncovered PoI p_s closest to u.
      if the SDC T_s and the TDC T_{tr} are satisfied **then**
        Insert PoI p_s into L_m and put PoI p_s into V_c
      end if
    end if
  end for
  $m \leftarrow m + 1$
end while
return m and L_1,L_2,...,L_m.

will return the mobile node to the sink node to hand over the previously collected data. Then, the mobile node can use the remaining sensing time to cover other PoIs. As depicted in Fig. 2(b), after adding PoI p_3 to the current path, assume that no more PoIs can be added to the current path directly due to the TDCs of some PoIs. However, if the mobile node has enough remaining sensing time after returning to the sink node, we can still add PoI p_4 into current mobile node's movement path.

After the mobile node passing through the sink node, the residual SDC T_s of each PoI is changed to $T_s = T_s - \frac{L}{v}$, and the TDCs of PoIs on the previous movement path are satisfied. Suppose that the path of the ith time a mobile node passing through the sink node is represented by $cycle_i$, at last, the movement path of a mobile node is the sum of these paths:

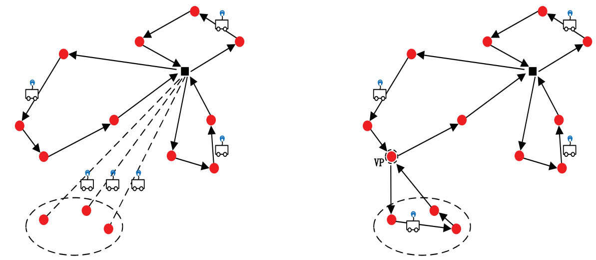
$$Path = cycle_1 + cycle_2 + ... + cycle_m. \tag{3}$$

Suppose OPT is the number of mobile nodes required in the optimal solution. The movement path of mobile nodes in the optimal solution is $PATH^* = \{Path_1^*, Path_2^*, ..., Path_{OPT}^*\}$. We use $C(Path_i)$ to denote the length of $Path_i$. The total length of the movement path of all mobile nodes is denoted by $C(PATH)$, thus in the optimal solution, we can get $C(PATH^*) = \sum_{i=1}^{OPT} C(Path_i)$. In the MR-MinExpand algorithm, for each mobile node in the MR-MinExpand algorithm, we have $C(Path_i) = \sum_{i=1}^{m} C(cycle_i)$. In the following, we will give the approximation ratio of the MR-MinExpand algorithm.



(a) (b)

Fig. 3: Illustration of obtaining a spanning tree.



(a) (b)

Fig. 4: Illustration of CoTSweep algorithm.

Theorem 2. *MR-MinExpand algorithm finds a solution with the number of mobile nodes required at most* 2ρ *OPT, where* $\rho = \frac{max(T_{tr},T_s)}{min(T_{tr},T_s)}$ *is the ratio of the maximum delay constraint to the minimum delay constraint.*

Proof. In timely sweep coverage, each mobile node starts from the sink node and returns to the sink node after covering a certain number of PoIs. As shown in Fig. 3(a) and Fig. 3(b), remove one of the two paths that connect to the sink node on the movement path to yield a tree for each mobile node. Since all mobile nodes start from the sink node, we obtain a spanning tree with the sink node as the root. Suppose MST is the minimum spanning tree of the graph induced by the set of PoIs and the sink node. Hence, the total path length of the optimal solution is greater than the edge length of the MST, i.e. $C(PATH^*) >= C(MST)$.

According to the step 4 to step 13 of the MR-MinExpand algorithm, it uses a greedy method to generate a mobile node's movement path. We use MST_i to denote the corresponding minimum spanning tree of the movement path of a mobile node i. For each mobile node i, we have $C(Path_i) <= 2C(MST_i)$. Suppose the output of the MR-MinExpand algorithm is N, then the overall path length $C(PATH) = \sum_{i=1}^{N} C(Path_i)$. Therefore, the total length of the mobile nodes' movement paths in the MR-MinExpand algorithm is upper bounded by $2C(MST)$. Hence, we can get $C(PATH) <= 2C(PATH^*)$.

We use d^* and d to denote the average length of the movement path in the optimal solution and the MR-MinExpand algorithm, respectively. Thus, we have

$$N \leq \frac{C(PATH)}{d} \leq \frac{2C(PATH^*)}{d} \leq \frac{2C(PATH^*)}{d^*} \cdot \frac{d^*}{d}$$
$$\leq 2 \cdot OPT \cdot \frac{d^*}{d} \leq 2 \cdot \frac{max(T_{tr},T_s)}{min(T_{tr},T_s)} \cdot OPT.$$

Let $\rho = \frac{max(T_{tr},T_s)}{min(T_{tr},T_s)}$ is the ratio of the maximum delay constraint to the minimum delay constraint. Thus, the approximate ratio of the MR-MinExpand algorithm is $\frac{N}{OPT} \leq 2\rho$. □

In the MR-MinExpand algorithm, note that we need to find the candidate PoI for each edge, which has the time complex of $O(n^2)$. At step 9 of the algorithm, the operation can be done in $O(n)$. Since in each iteration, a PoI is picked for the mobile node, the overall time complexity is $O(n(n^2+n))$. Hence, the time complexity of the MR-MinExpand algorithm is $O(n^3)$.

V. Cooperative Timely Sweep Coverage (CoTSweep)

When using the MR-MinExpand algorithm to solve the MNTSC problem, the following issues may exist. Since some PoIs are far from the sink node, a mobile node may only cover one PoI on its movement path, or some of remote PoIs cannot be covered by any mobile node. Therefore, we propose an algorithm called CoTSweep for these issues.

Usually, in order to submit data to the sink node, mobile nodes need to periodically return to the sink node. As shown in Fig. 4(a), some PoIs are far from the sink node. In order to using mobile nodes to cover those PoIs, as shown in Fig. 4(b), we transmit the collected data back to the sink node through the collaboration between mobile nodes. Apparently, in this way, some mobile nodes do not need to return to the sink node, thus they have more time to cover remote PoIs. In the CoTSweep algorithm, we will use this kind of collaboration between mobile nodes to cover PoIs with fewer mobile nodes.

The CoTSweep algorithm is shown in Algorithm 2. According to the algorithm, some previously added mobile nodes will be selected to help transmit data from remote PoIs. Therefore, the mobile nodes covering remote PoIs do not need to return to the sink node for data transmission by themselves. We assume that when two mobile nodes move to the same position at the same time, the data can be transmitted.

Extra energy may be required to complete the collaboration between mobile nodes. However, this can be solved by using a larger capacity battery on the mobile node or arranging some wireless chargers in the sensing field [17]. In addition, in the MNTSC problem, how to cover the PoIs with the minimum number of mobile nodes is our most concerned. Therefore, we assume that mobile nodes have enough energy in their working process, and how to achieve timely sweep coverage under the restriction on the energy consumption of mobile nodes is the problem we will study in future.

It is feasible to transmit data through this kind of collaboration between mobile nodes. Although the speed of the mobile node is v, we can adjust the average speed of a mobile node by stopping the mobile node for a period of time. Suppose that the average speed of a mobile node on its movement path is denoted by $\overline{v}$. Therefore, we can fine tune the starting time and average speed $\overline{v}$ of the mobile node to realize the collaboration between mobile nodes.

In fact, the position of data delivering between two mobile nodes may be any point on the mobile node's movement path.

Algorithm 2 Cooperative Timely Sweep Coverage, CoTSWeep

Input: The PoI set V, the sink node u, the SDC T_s, and the TDC T_{tr}.
Output: The number of mobile nodes m required and the sweep paths L_1,L_2,...,L_m.

1: Using the MR-MinExpand algorithm to construct the sweep path set $L = \{L_1, L_2, ..., L_k\}$.
2: Set $m \leftarrow k$, and update the already covered PoI set V_c.
3: **for** $L_i \in L$ **do**
4:     **if** L_i contains only one PoI **then**
5:         remove the PoI from V_c, and delete L_i from L.
6:     **end if**
7: **end for**
8: **while** $V \neq V_c$ **do**
9:     **for** $L_i \in L$ **do**
10:         Calculate the VP vp_i.
11:     **end for**
12:     Choose the most cost-effective VP and denote it as vp_p.
13:     Using the MR-MinExpand algorithm to construct the movement path in which vp_p is the chosen VP.
14:     Update m, V_c and path set L.
15:     **if** V_c and L are unchanged **then**
16:         break;
17:     **end if**
18: **end while**
19: **return** m and L_1,L_2,...,L_m.

We call this position a *virtual point* (*VP*). For the sake of simplicity, in our paper, we assume that the position of a VP is selected from the positions of PoIs on the movement path. In the CoTSweep algorithm, the position of a VP is found by choosing the most cost-effective position among the positions of the already covered PoIs, i.e.,

$$max(\frac{N^{'}}{M^{'}}), \tag{4}$$

where $N^{'}$ is the number of PoIs that can be newly covered, $M^{'}$ is the number of mobile nodes required for covering the remote PoIs. The CoTSweep algorithm selects the most cost-effective VP vp_p and updates the set of the covered PoI V_c. If some PoIs still cannot be covered, the above procedure is repeated to choose a new VP until all the PoIs are covered or no more PoIs can be added to V_c.

When calculating the position of a VP, if we take every covered PoI as a candidate VP, the computation of the algorithm will be very high. Therefore, we use the following method. For the PoIs on each generated mobile node's movement path, we only select one PoI as the candidate VP. Suppose for each generated mobile node's movement path, the nearest PoI from the remaining uncovered PoIs is selected and represented by p_t. The sink node is u and $Path_i$ is the movement path of mobile node i. The position of a VP on a movement path is calculated by solving the following problem, which aims to give a mobile node more remaining time to cover other PoIs:

$$\textbf{Maximum} \quad \min(T_{tr} - \frac{len(vp_k^i, u)}{v}, T_s) - \frac{dist(p_t, vp_k^i)}{v}$$
$$\textbf{s.t.} \quad vp_k^i \in Path_i; \quad len(vp_k^i, u) \leq v * T_{tr}; \quad 2 * dist(p_t, vp_k^i) \leq T_s * v.$$

In step 12 of the CoTSweep algorithm, suppose a VP vp_i is selected on $Path_i$ and the remaining time is T_R, then CoTSweep uses the MR-MinExpand algorithm to cover the remaining PoIs until some PoIs' constraints are not met. The above process will be repeated to find a new VP.

As shown in Algorithm 2, the first part of CoTSweep uses the MR-MinExpand algorithm to cover PoIs. The second part leverages mobile node collaboration to cover the remaining PoIs. The time complexity of the MR-MinExpand algorithm is $O(n^3)$. Therefore, the time complexity of the first part of the CoTSweep algorithm is $O(n^3)$. The running time of the second part of the CoTSweep algorithm is related to the number of PoIs that are not covered by the first part of the algorithm. Assume that the number of uncovered PoIs is denoted by W. It is easy to get the fact that W is smaller than the number of PoIs whose distance from the sink node is greater than $min\{T_{tr} \cdot v, \frac{T_s}{2} \cdot v\}$. The time complexity of the CoTSweep algorithm's step 9 to step 11 is $O(n^2)$. The time complexity of step 12 and step 13 is $O(n^3)$. Hence, the complexity of the second part of CoTSweep is $O(W(n^2+n^3+n^3))$. Therefore, the overall complexity of CoTSweep is $O(Wn^3)$.

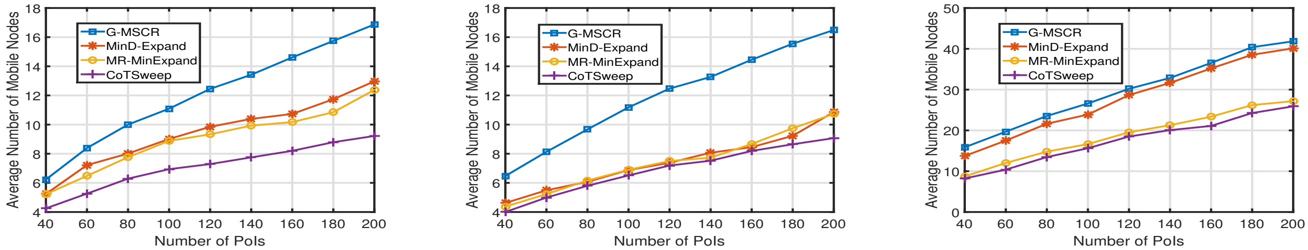
VI. Performance evaluation

A. simulation setup

In our simulation, we randomly deploy a number of PoIs on a 2-D plane. The nodes' positions are uniformly distributed. The area of the plane is 200m by 200m and the movement velocity of the mobile node is 1m/s. We compare our proposed algorithms MR-MinExpand and CoTSweep with G-MSCR and MinD-Expand [8]. We conduct the simulation under two different scenarios. One is all PoIs can be covered by mobile nodes under the delay constraints. The other is that some remote PoIs cannot be covered under the delay constraints by a single mobile node. We run each simulation for 100 times, and get the average results under each scenario.

B. The number of required nodes

The simulation results on the number of mobile nodes required are summarized in Fig. 5. We can find out that the CoTSweep algorithm requires the minimum number of mobile nodes and the G-MSCR algorithm requires the maximum number of mobile nodes. In Fig. 5(a) and Fig. 5(b), we can also find that MR-MinExpand and MinD-Expand have similar performance. Because when $T_{tr} \geq T_s$, MR-MinExpand and MinD-Expand have the similar strategy of selecting the uncovered PoI. Fig. 5(c) shows the case when $T_{tr} < T_s$. Apparently MR-MinExpand and CoTSweep outperform G-MSCR and MinD-Expand. It is because, in MR-MinExpand and CoTSweep, when $T_{tr} < T_s$, the movement paths of



(a) T_{tr}=400s, T_s=400s (b) T_{tr}=600s, T_s=400s (c) T_{tr}=150s, T_s=400s

Fig. 5: Average number of mobile nodes required.

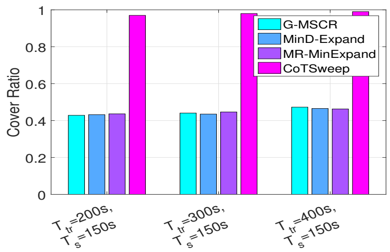


Fig. 6: Comparison of the cover ratio.

mobile nodes pass through the sink node more than once, and the mobile node can submit the previously collected data when passing through the sink node. Moreover, CoTSweep has the better performance than MR-MinExpand. The reason is that the CoTSweep algorithm utilizes the collaboration between mobile nodes, so it can handle the situation when a mobile node only cover a PoI, which makes the scheduling scheme of CoTSweep more cost-effective.

C. The ability to cover remote PoIs

The cover ratio is the ratio of the number of covered PoIs to the total number of PoIs in the field. For a sink node, an algorithm with high cover ratio can make it get data from more PoIs. Fig. 6 shows the simulation results of the four algorithms under different delay constraints. We can easily find that the CoTSweep algorithm outperforms the other three algorithms. The reason for this is that in CoTSweep, the remote PoIs can be covered through the collaboration between mobile nodes, while other algorithms that do not use the collaboration between mobile nodes can not cover those remote PoIs.

VII. Conclusion

In this paper, we investigate the MNTSC problem, which aims to find the minimum number of mobile nodes required to achieve timely sweep coverage of a sensing field. We propose two heuristic algorithms, MR-MinExpand and CoTSweep, to solve the MNTSC problem. In these two algorithms, by exploiting the difference between SDC and TDC, the mobile node can cover more PoIs, so the algorithms require fewer mobile nodes. Furthermore, the CoTSweep algorithm can solve the problem that some remote PoIs cannot be covered by a single mobile node through the use of the collaboration between mobile nodes. The simulation results show that the two proposed algorithms can cover PoIs in the sensing field with fewer mobile nodes and CoTSweep has advantages in covering remote PoIs.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under No. 61732017, the Funds for International Cooperation and Exchange of NSFC under Grant No. 61720106007, and the 111 Project (B18008).

References

[1] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, 2006.

[2] Huadong Ma, Dong Zhao, and Peiyan Yuan. Opportunities in mobile crowd sensing. *IEEE Communications Magazine*, 52(8):29–35, 2014.

[3] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang, and Xingshe Zhou. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Computing Surveys*, 48(1):1–31, 2015.

[4] Dong Zhao, Huadong Ma, and Liang Liu. Mobile sensor scheduling for timely sweep coverage. In *Proc. of IEEE WCNC, 2012*.

[5] Li Shu, Ke Wei Cheng, Xiao Wen Zhang, and Ji Liu Zhou. Periodic sweep coverage scheme based on periodic vehicle routing problem. *Journal of Networks*, 9(3):726–732, 2014.

[6] Mo Li, Weifang Cheng, Kebin Liu, Yuan He, Xiangyang Li, and Xiangke Liao. Sweep coverage with mobile sensors. *IEEE Transactions on Mobile Computing*, 10(11):1534–1545, 2011.

[7] Junzhao Du, Yawei Li, Hui Liu, and Kewei Sha. On sweep coverage with minimum mobile sensors. In *Proc. of IEEE ICPADS, 2010*.

[8] Chuang Liu, Hongwei Du, and Qiang Ye. Sweep coverage with return time constraint. In *Proc. of IEEE GLOBECOM, 2016*.

[9] Zhiyin Chen, Xudong Zhu, Xiaofeng Gao, Fan Wu, Jian Gu, and Guihai Chen. Efficient scheduling strategies for mobile sensors in sweep coverage problem. In *Proc. of IEEE SECON, 2016*.

[10] Yuchen Feng, Xiaofeng Gao, Fan Wu, and Guihai Chen. Shorten the trajectory of mobile sensors in sweep coverage problem. In *Proc. of IEEE GLOBECOM, 2015*.

[11] Chuang Liu, Hongwei Du, and Qiang Ye. Utilizing communication range to shorten the route of sweep coverage. In *Proc. of IEEE ICC, 2017*.

[12] Xiaofeng Gao, Jiahao Fan, Fan Wu, and Guihai Chen. Approximation algorithms for sweep coverage problem with multiple mobile sensors. *IEEE/ACM Transactions on Networking*, 26(99):1–14, 2018.

[13] Xiaofeng Gao, Xudong Zhu, Yuchen Feng, Fan Wu, and Guihai Chen. Data ferry trajectory planning for sweep coverage problem with multiple mobile sensors. In *Proc. of IEEE SECON, 2016*.

[14] Hung-Chi Chu, Wei-Kai Wang, and Yong-Hsun Lai. Sweep coverage mechanism for wireless sensor networks with approximate patrol times. In *Proc. of IEEE UIC-ATC, 2010*.

[15] Weifang Cheng, Mo Li, Kebin Liu, Yunhao Liu, Xiangyang Li, and Xiangke Liao. Sweep coverage with mobile sensors. In *Proc. of IEEE IPDPS, 2008*.

[16] Meng Yang, Donghyun Kim, Deying Li, Wenping Chen, Hongwei Du, and Alade O Tokuta. Sweep-coverage with energy-restricted mobile wireless sensor nodes. In *Proc. of Springer WASA, 2013*.

[17] Zhi Ma, Jie Wu, Sheng Zhang, and Sanglu Lu. Prolonging wsn lifetime with an actual charging model. In *Proc. of IEEE WCNC, 2018*.