Approximation Algorithms for Sweep Coverage Problem With Multiple Mobile Sensors

Xiaofeng Gao, *Member, IEEE*, Jiahao Fan, Fan Wu, *Member, IEEE*, and Guihai Chen, *Senior Member, IEEE*

***Abstract*— Sweep coverage plays an important role in many applications like data gathering, sensing coverage, and devices control. In this paper, we deal with the sweep coverage problem with multiple mobile sensors to periodically cover n targets in the surveillance region. We propose three constant-factor approximations, namely, *CycleSplit*, *HeteroCycleSplit*, and *PathSplit*, to minimize the longest sweep period of mobile sensors under different scenarios, respectively. *CycleSplit* deals with the min-period sweep coverage problem (MPSC), in which each mobile sensor works independently along a predetermined trajectory cycle. It has an approximation ratio of $(5 - 2/(n - m + 1))$, which improves the best known approximation ratio of 5. *HeteroCycleSplit* is a 5α-approximation. It computes the sensor routes for heterogeneous velocity min-period sweep coverage problem (HVMPSC), where each mobile sensor has a different velocity. *PathSplit* is a 2-approximation for connected path min-period sweep coverage problem (CPMPSC). It solves a variant problem of sweep coverage where we need to cover all the given edges. Besides, we also propose an optimal algorithm *DP-MPSC* for min-period sweep coverage problem in 1-D case. Finally, we provide various numerical experiments and comparisons with several previous work to validate the efficiency of our design.**

***Index Terms*— Sweep coverage, mobile sensor, TSP cycle, approximation.**

I. Introduction

COVERAGE problems in Wireless Sensor Networks (WSN) have been studied extensively under various models. Briefly speaking, this kind of problems requires a strategy of deploying wireless sensors to collect useful information about a given region. Some studies focus on covering PoIs (Points of Interest), and formulate the problem as *Target Coverage* problem [1]. In target coverage problem, we need to use a number of wireless sensors to cover a set of static targets. Besides, some consider *Area Coverage* problem, in which the objective is to fully cover or partially cover the given area with a given number of static wireless sensor. This is a traditional problem about sensor coverage and has been used in many practical scenarios. Another interesting application is *Barrier Coverage*: we want to monitor a belt region, such that any intruder that tries to cross the region will be detected [2]–[4]. Researchers also investigate the coverage issues for directional sensor networks [5] called *Camera Sensor Coverage*. An object in camera sensor coverage is often called full-view covered if there is always a camera sensor facing close to it no matter which direction it faces [6].

Instead of continuous monitoring, many applications only require periodic patrol inspection for a certain set of PoIs. Typical examples include police patrolling, message ferrying, devices control, etc. In such scenario, a mobile sensor can move around to collect data from targets actively, and the objective is usually to minimize the number of detecting sensors under a time constraint or find the minimum sweep period given the number of targets or shorten the trajectory length of mobile sensors. We refer to such problem as *Sweep Coverage* [7]–[11]. Similar models have also been studied under the context of autonomous robots, vehicle routing, and data collection.

In this paper, we mainly focus on sweep coverage problem with multiple mobile sensors. Assume that there are n targets in the surveillance region and m mobile sensors. Each mobile sensor works as a data ferry to collect information from targets. If a sensor moves to the position of a target, then this target is considered to be detected by the mobile sensor. Imagine that all the mobile sensors move along a set of predefined trajectories continuously to collect data, and a target is said to be t-sweep covered if it is detected by a mobile sensor at least once every t time units (we call t its sweep period). The objective here is to minimize the sweep period for all targets. We consider three variations of this problem. Their detailed descriptions are as follows.

To begin with, assume that all mobile sensors have the same velocity v. In *Min-Period Sweep Coverage* problem (MPSC), each mobile sensor moves along one cycle and each target belongs to exactly one of these cycles. An example is shown in Figure 1. The sweep period of the target on cycle $\mathcal{C}$ is $\frac{Len(\mathcal{C})}{v}$, which is proportional to $l = Len(\mathcal{C})$. Here $Len(\mathcal{C})$ denotes the length of cycle $\mathcal{C}$. If we assign l as the trajectory length for the sensor on $\mathcal{C}$, then minimizing the sweep period for PoIs is equal to minimizing the maximum trajectory length for all mobile sensors.

We further consider a more realistic version of Min-Period Sweep Coverage, in which the speeds of sensors are heterogeneous. In other words, the sweep period for each target on cycle $\mathcal{C}$ covered by the i^{th} mobile sensor is $\frac{Len(\mathcal{C})}{v_i}$. We change

Manuscript received August 7, 2017; revised January 19, 2018; accepted March 2, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P.-J. Wan. Date of publication April 2, 2018; date of current version April 16, 2018. This work was supported in part by the Program of International S&T Cooperation under 2016YFE0100300, in part by the China 973 Project under Grant 2014CB340303, in part by the National Natural Science Foundation of China under Grant 61472252, Grant 61672353, and Grant 61672348, in part by the Shanghai Science and Technology Fund under Grand 17510740200 and Grant 15220721300, and in part by the CCF-Tencent Open Research Fund under Grant RAGR20170114. *(Corresponding author: Xiaofeng Gao.)*

The authors are with the Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: gao-xf@cs.sjtu.edu.cn; j.h.fan@sjtu.edu.cn; fwu@cs.sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

Digital Object Identifier 10.1109/TNET.2018.2815630

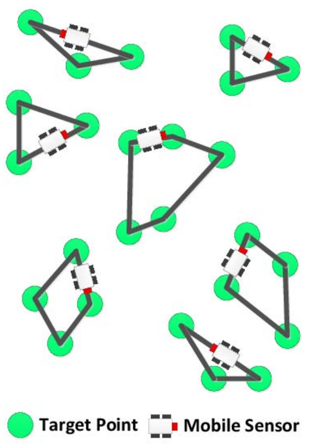

Fig. 1. An illustration for sweep coverage problem.

the expression to $\mathcal{W}_i Len(\mathcal{C})$ and assign it as the weighted trajectory length for the sensor on $\mathcal{C}$. The goal is to minimize the maximum weighted trajectory length so as to minimize the sweep period for all n targets. We refer to it as *Heterogeneous Velocity Min-Period Sweep Coverage* problem (HVMPSC).

There is also another version of Min-Period Sweep Coverage. Here we want to cover not only the targets, but also some connected edges. This is useful when periodic patrol inspection is required for several routes or continuous paths. Given a set of connected edges E_s and m mobile sensors, we are supposed to schedule the path for each mobile sensor to cover all the edges that belong to E_s. We refer to this problem as *Connected Path Min-Period Sweep Coverage* problem (CPMPSC), the sweep period of which is $\frac{Len(P_i)}{v}$. Again, we use the trajectory length $Len(P_i)$ in the objective of this problem.

Easy to see, if $m = 1$, the Min-Period Sweep Coverage problem becomes a Traveling-Salesman-Problem (TSP), which is a classic NP-complete problem. Therefore MPSC, HVMPSC and CPMPSC are all NP-complete. In previous literature, the MPSC problem is also known as Min-Max Cycle Cover Problem (MMCCP) [12], which looks for at most k disjoint cycles to cover the given targets with a minimum cost. Correspondingly, Yu and Liu [12] proposed a 5-approximation to solve this problem, which is the best approximation up to now. Besides, [13] studied the *k-TSP* problem which is similar to MPSC. However, the cycles in [13] share a common starting point while we use separated cycles in MPSC. There is an approximation in [14] for Heterogeneous Velocity Min-Period Sweep Coverage problem. However, when the range between the groups of targets are too large, its approximation of output cannot maintain. There has been no approximation to solve Connected Path Min-Period Sweep Coverage problem yet.

Correspondingly, in this paper we propose three constant-factor approximations, namely *CycleSplit*, and *HeteroCycleSplit*, *PathSplit* to solve the above mentioned three variations respectively. The first approximation, *CycleSplit*, deals with MPSC problem with the approximation ratio of $(5 - \frac{2}{n-m+1})$. It improves the best known approximation ratio of 5 in [12]. The second approximation *HeteroCycleSplit* is a 5α-approximation for the Heterogeneous Velocity Min-Period Sweep Coverage problem. It implements the modified CycleSplit algorithm to achieve the goal. The third approximation is a 2-approximation called *PathSplit* for Connected Path Min-Period Sweep Coverage problem. Besides, we also provide an optimal algorithm for Min-Period Sweep Coverage problem in one dimensional case.

We relax our discussion to metric space since we only use the triangle inequality property in our analysis. Thus, in the following discussion, all the problems are defined in metric space unless explicitly mentioned, which means that our algorithms are applicable to more general cases.

Next, we provide various numerical experiments to validate the efficiency of our design. We compare our algorithms with OSweep algorithm in [15], MinExpand algorithm proposed in [15] and PDBA algorithm in [16]. The performance evaluation shows that our algorithms can achieve better results under the same network scale.

To sum up, the contributions of our paper are as follows.

- we consider three variations of sweep coverage problem under metric space with multiple mobile sensors, namely *Min-Period Sweep Coverage* problem (MPSC), *Heterogeneous Velocity Min-Period Sweep Coverage* (HVMPSC) problem and *Connected Path Min-Period Sweep Coverage* (CPMPSC) problem. The objective is to shorten the longest trajectory length of mobile sensors to reduce the target detection period.
- We propose one optimal algorithm, DP-MPSC for Min-Period Sweep Coverage problem in one dimensional case.
- We propose three constant-factor approximations *CycleSplit*, *HeteroCycleSplit* and *PathSplit* to solve these three problems respectively. Both CycleSplit and HeteroCycleSplit have better approximation ratio than the best state-of-the-art algorithms, while PathSplit is the first constant-factor approximation for Sweep Coverage problem in this scenario.
- We compare our algorithms with several previous literatures by simulations. Both theoretical analysis and numerical experiments validate the efficiency of our design.

The rest of this paper is organized as follows. Section II discusses some related work. Section III introduces the preliminaries for later sections. In Section IV, Section V and Section VI, we propose algorithms for Min-Period Sweep Coverage, Heterogeneous Velocity Min-Period Sweep Coverage and Connected Path Min-Period Sweep Coverage respectively with approximation analysis. In Section VII, we give experiments to evaluate the performance of our algorithms. Section VIII is the final conclusion and future work.

II. Related Work

The min-period sweep coverage problem with multiple sensors is closely related to multiple Traveling-Salesman-Problem (TSP) with min-max objective (denoted as *min-max m-TSP* problem, or *min-max k-cycle* problem). The traditional single TSP is one of the most intensively studied problems in the area of combinational optimization. A simple algorithm based on minimum spanning tree (MST) gives a 2-approximation solution. By a clever construction,

TABLE I
SOME PREVIOUS WORK CONCERNING SWEEP COVERAGE PROBLEM

Paper	Problem description	Algorithm name	Approximation ratio
[10]	Achieve global t-sweep coverage with minimum number of sensors	CSWEEP	2+ϵ
		GSWEEP	3
	Locally decide the moving path based on the knowledge exchanged with other sensors	DSWEEP	
[14]	Minimize the maximum inter-arrival time of a data ferry to collect data from a sensor node		$1.5 \cdot 2\pi \cdot 2\alpha \cdot (1 + 20/\pi)$
[15]	Achieve global t-sweep coverage with minimum mobile sensors	MinExpand	
[16]	Minimize the number of mobile sensors for sweep coverage	PDBA	heuristic
[17]	Find the minimum number of mobile sensors to guarantee sweep coverage of vertices of a graph	GSWEEPCOVERAGE	3
[18]	Minimize the makespan of mobile sweep routes	BS-Sweep	6
[19]	The minimum number of required sensors problem in sweep coverage and the vehicle routing problem	VRPSC Scheme	heuristic
[20]	Minimize the average delay in delivering data using mobile elements in wireless networks	PSH	heuristic
[21]	Determine the minimum velocity for a single mobile sensor to satisfy the double delay constraints	STSP	$2\max(T_s, T_{tr})/\min(T_s, T_{tr})$
[22]	Consider the impact of sensing range and find the shortest path on which a mobile sensor can sweep around the targets under the delay constraint	G-ROSE	$1 + \rho$
[23]	Sweep coverage for covering a set of line segments on a plane	LINESWEEPCOVERAGE	2
	Minimize the number of data mules to periodically collect data from a set of mobile sensor nodes	MDMDG	3
[24]	Point sweep coverage problem	POINTSWEEPCOVERAGE	2
	Area sweep coverage problem	AREASWEEPCOVERAGE	$2(\sqrt{2} + \frac{2r\mathcal{P}}{Area(\mathcal{A})})$
[25]	Find the minimum number of mobile sensor nodes to cover the POIs and deliver the collected data to the base station within a preset time window	G-MSCR	heuristic
		MinD-Expand	heuristic
[26]	Minimize the number of sensors considering periodical coverage of POIs and delivery of data simultaneously	ACOSC	heuristic

Christofides [27] improved the approximation ratio from 2 to 1.5. It has been proved that the metric TSP is inapproximable within a ratio of $\frac{123}{122}$, unless P $=$ NP [28]. Obviously, the min-period sweep coverage problem (MPSC) is at least as hard as the original TSP.

Following the similar strategy, we could firstly find a min-max k-tree for a given graph, and then construct cycles based on these trees. If the min-max k-tree problem has an α-approximation, then there is a 2α-approximation for min-max k-cycle problem. Unfortunately, min-max k-tree is also an NP-hard problem, which is proved in [29] by reduction from Bin Packing. Even *et al.* [29] presented a 4-approximation algorithm for min-max k-tree and a 4-approximation algorithm for its rooted variant. In 2014, Khani and Salavatipour [30] improved the approximation ration from 4 to 3. It has also been proved in [31] that the min-max k-tree problem and its rooted variant have an inapproximability bound of $\frac{3}{2}$. Therefore, following this approach, we cannot design an approximation with ratio better than 3 to min-max k-cycle problem.

Researchers have made many contributions to solve problems concerning sweep coverage [10], [14]–[26], [32]–[42]. Related literatures usually focused on these problems: finding the minimum sweep period or determining the minimum number of mobile sensors. Table I shows some previous work concerning these problems.

Li *et al.* [10] proposed two centralized algorithm for scenarios about Global t-Sweep Coverage. The first centralized algorithm named CSWEEP aims to compute a globe TSP cycle based on the positions of PoIs and then split it into several distinct loops for mobile sensors. The second centralized algorithm named GSWEEP considers that different PoIs have different sweep period and replaces each original PoI with several virtual PoIs. Then GSWEEP converts the new problem to the original problem and solves it with CSWEEP. For practicability and scalability, they proposed a distributed sweep algorithm, DSWEEP, which cooperates sensors efficiently to provide required coverage with the best effort.

Xue *et al.* [14] proposed several new variations of sweep coverage problem with neighborhood. An approximation algorithm is introduced in this paper to solve the sweep coverage problem called k-ITSPN without a fixed initial velocity. However this algorithm cannot be used in all scenarios. When the range between the groups of targets are too large, the approximation of output cannot maintain.

Du *et al.* [15] designed a centralized algorithm named MinExpand to find the minimum number of mobile sensors required for sweep coverage. MinExpand gradually deploys more mobile sensors and schedules distinct sweep route for new mobile sensor according to the positions of the remaining PoIs and sweep period t until all PoIs satisfy t-sweep coverage requirement.

Liu *et al.* [16] proposed a heuristic, called the perpendicular-distance-based algorithm (PDBA), to minimize the number of mobile sensors and the total energy consumed. They chose a PoI to join the route of a mobile sensor according to its perpendicular distance to the bottom of the graph. The algorithm deploys more mobile sensors when the paths of the existing mobile sensors exceeds the length constraint.

Gorain and Mandal [17] considered the sweep coverage problem to find the minimum number of mobile sensors to ensure periodic monitoring for a given set of points of interest. They proposed a 3-approximation algorithm named GSWEEPCOVERAGE for this problem.

Feng [18] proposed a constant-factor approximation BS-Sweep for the optimization problem to minimize the makespan of mobile sweep routes (M^{3}SR) based on the idea of binary search. They use the weight of the minimum spanning forests to continuously check if the expected value can be achieved with the given conditions.

Li *et al.* [19] would like to support dynamical POI coverage and data delivery simultaneously by modeling the minimum number of required sensors problem in sweep coverage as a Vehicle Routing Problem (VRP). They proposed a novel sweep coverage scheme, named VRPSC (Vehicle Routing Problem based Sweep Coverage) to address the problem. An greedy insertion algorithm is designed to create the initial scanning routes for POIs, and then the Simulated Annealing is employed to optimize these routes.

Moazzez-Estanjini and Paschalidis [20] presented an algorithm named PSH to minimize the average delay in delivering data using mobile elements in wireless networks. A single sink is located among the POIs to collect data from multiple mobile sensors. PSH transforms Hamiltonian solutions to potentially better non-Hamiltonian solutions by splitting an available Hamiltonian solution into several loops.

Zhao *et al.* [21] presented a new problem in which every mobile sensor not only covers the PoIs but also visits a base station periodically. They considered the Double Delay Constrained Min-Velocity Scheduling (DDC-MVS) problem, to determine the minimum velocity for a single mobile sensor to satisfy the double delay constraints. They proposed two algorithms STSP and ITSP to solve the problem. The solution of STSP can be very bad when the two delay constraints differ greatly. So they designed ITSP to improve the solution.

Chen *et al.* [22] considered the impact of sensing range and proposed the Distance-Sensitive-Route-Scheduling (DSRS) problem. In this scenario, each sensor has its sensing range, and whenever a target falls into the sensing area, it is considered to be detected. They designed an approximation called G-ROSE to address the route scheduling problem.

Gorain and Mandal [23] wanted to cover a set of line segments on a plane. They designed a 2-approximation algorithm to find a solution. They also proposed a 3-approximation algorithm for periodically data gathering from mobile sensor nodes with minimum number of data mules. The main difference between their line sweep coverage problem and our connected path min-period sweep coverage problem (CPMPSC) is that the edges in their problem are all separated from each other, while the edges in CPMPSC belong to a connected graph according to our definition.

Gorain and Mandal [24] discussed the Point Sweep Coverage Problem, to minimize the number of mobile sensor nodes with a constant velocity. They proposed an 2-approximation algorithm to solve the point sweep coverage problem when the PoIs are static sensor nodes. They also introduced a new problem called Area Sweep Coverage and provided an approximation to solve it.

Liu *et al.* [25] studied the problem of sweep coverage with return time constraint. This problem requires that the POIs should be covered and the collected data should be delivered to the base station within a preset time window. G-MSCR and MinD-Expand are two heuristic algorithms designed to solve this problem. In practice, G-MSCR leads to shorter return time and MinD-Expand requires fewer sensor nodes.

Huang *et al.* [26] considered the data delivery to sink in sweep coverage and designed the ACOSC algorithm based on the ant colony optimization (ACO) algorithm to search for a solution. In ACOSC, each ant builds a list of feasible movements and chooses the one by a probabilistic rule. The total mobile sensor number is minimized to decrease the network cost.

III. Preliminary

In this section, we define some basic concepts and introduce some primitive methods which will be used in later sections.

A. Metric Space

For any given complete graph $G(V, E)$, we will use d to represent a metric on V such that $d : V \times V \to \mathbb{R}^+$. Triangle inequality is the most important property that a metric space holds, i.e. $d(x, z) \leq d(x, y) + d(y, z)$ for any $x, y, z \in V$.

Suppose e is the edge between vertices u and v, we will use $d(u, v)$ and $d(e)$ to denote the same thing in context without ambiguity, which represents the distance between points u and v or the length of edge e. For an edge set $E' \subset E$, define $d(E') = \sum_{e \in E'} d(e)$.

B. Cycle Cover and Tree Cover

The formal definition of Cycle Cover is shown as follows.

Definition 1 (Cycle Cover): Given a graph $G = (V, E)$ and a vertex set $V' \subseteq V$, a cycle cover for V' is a set of cycles $\mathscr{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_t\}$, which are subgraphs of G, and the union of their vertices is V'.

Similarly, we can define tree cover as follows.

Definition 2 (Tree Cover): Given $G = (V, E)$ and $V' \subseteq V$, a tree cover for V' is a forest $\mathscr{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_t\}$, which are subgraphs of G, and the union of their vertices is V'.

C. Constructing Cycle From Tree

We could use Christofides' improvements [27] to construct cycles from trees in our algorithm, but we cannot trivially get a better approximation ratio. The approximation ratio of Christofides' algorithm is 1.5. We briefly explain how Christofides' algorithm works. Let $G(V, E)$ be an instance of the Travelling Salesman Problem. In other words, G is a complete graph on the set V of vertices, and the set E assigns a non-negative real weight to every edge of G.

The whole algorithm for constructing cycle from tree can be described as follows.

1) Find a minimum spanning tree T of G.
2) Let E' be the set of vertices whose degree is odd in T. By the handshaking lemma, E' has an even number of vertices.
3) Find a minimum-weight perfect matching M in the induced subgraph given by the vertices from E'.
4) Combine the edges of M and T to form a connected multigraph G' in which every vertex has even degree.
5) Form a Eulerian circuit in G' and make the circuit found in previous step into a Hamiltonian circuit by removing the repeated vertices from the circuit, which is called shortcutting.

Besides, there is a useful theorem in [27], which will later be used in our own proof.

Theorem 1: A Hamiltonian circuit $\mathcal{C}_H$ of a complete graph G can be found with cost

$$C(\mathcal{C}_H) \leq C(\mathcal{T}^*) + \frac{1}{2}C(TSP^*) \leq \frac{3}{2}C(TSP^*),$$

where $\mathcal{T}^*$ is a minimum spanning tree of G and TSP^* is an optimal solution to the Travel Salesman Problem in G.

D. Global t-Sweep Coverage

Sweep coverage, unlike traditional area coverage or barrier coverage, does not require static and continuous coverage all the time. In sweep coverage, we only need to cover every PoI at least once every certain time interval to guarantee event detection within a certain delay bound. With this idea, we define *t-Sweep Coverage* as follows.

Definition 3 (t-Sweep Coverage): A PoI is said to be t-sweep covered by a coverage scheme $\mathcal{F}$ if and only if it is scanned at least once every t time units by the mobile sensors allocated by $\mathcal{F}$.

If a PoI is t-sweep covered, time interval t is called the *sweep period* of the PoI. When there is a set of PoIs, different PoIs may have different sweep periods. In order to unify the requirements, we need to define *Global t-Sweep Coverage* as follows.

Definition 4 (Global t-Sweep Coverage): A set of PoIs is said to be global t-sweep covered by a coverage scheme $\mathcal{F}$ if and only if all PoIs are scanned at least once every t time units by the mobile sensors allocated by $\mathcal{F}$.

E. Min-Sensor Sweep Coverage Problem

Some previous algorithms for sweep coverage problem are to find the minimum number of mobile sensors to satisfy the required global t-sweep coverage constraints for PoIs [10], [15], [16]. This problem is denoted as *Min-Sensor Sweep Coverage Problem* in [10]. Briefly, it is defined as follows.

Definition 5 (Min-Sensor Sweep Coverage Problem): Given input triple (G, T, d), where $G = (V, E)$ is a complete graph, T is the global sweep period constraint, $d : V \times V \rightarrow \mathbb{R}^+$ is a metric, Min-Sensor Sweep Coverage Problem aims to find a coverage scheme $\mathcal{F}$ such that the number of mobile sensors m is minimized.

Li *et al.* [10] showed 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, unless P = NP.

IV. Min-Period Sweep Coverage

In this section, we will formally define the Min-Period Sweep Coverage problem (MPSC), and then design an approximation named *CycleSplit* for this problem. We also propose an optimal algorithm for Min-Period Sweep Coverage problem in one dimensional case. Besides, we will prove the correctness of our algorithms.

A. MPSC: General Case

In this section, we will talk about Min-Period Sweep Coverage problem in general case. We firstly give the formal formulation of the problem and then propose an approximation for this problem, the approximation ratio of which is $(5 - \frac{2}{n-m+1})$.

1) Problem Formulation: The basic idea of our algorithms to solve the Min-Period Sweep Coverage problem follows the following procedures:

1) Construct a cycle cover $\mathscr{C}$;
2) Determine how many sensors will be allocated to each cycle in $\mathscr{C}$.
3) Split the cycle evenly and construct a new cycle for each mobile wireless sensor.

Mobile wireless sensors do not work cooperatively, and each sensor traverses a distinct cycle (as shown in Figure 1). We define it as *Min-Period Sweep Coverage* problem (MPSC).

Definition 6 (Min-Period Sweep Coverage (MPSC)): Given input triple (G, m, d), where $G = (V, E)$ is a complete graph, m is the number of sensors, $d : V \times V \rightarrow \mathbb{R}^+$ is a metric, Min-Period Sweep Coverage (MPSC) problem aims to find a cycle cover $\mathscr{C}$ such that $|\mathscr{C}| = m$ and $\max\limits_{1\leq i\leq|\mathscr{C}|} d(\mathcal{C}_i)$ is minimized.

As mentioned in Section I, MPSC problem is actually the multiple Traveling-Salesman-Problem with min-max objective (min-max m-TSP).

2) CycleSplit (An Approximation for MPSC): In this subsection we design *CycleSplit* algorithm to solve the MPSC problem. The main idea is to first select some connected components from the given graph, compute a TSP cycle for each connected component, split the TSP cycle into several segments and finally form distinct cycles.

Recall Kruskal's algorithm for constructing a minimum spanning tree. We add edges to the empty graph $G' = (V, \emptyset)$

one by one, by an increasing order of their length (i.e. $d(\cdot)$). In each stage of Kruskal's algorithm, G' will have a number of connected components, and each subgraph induced by the connected component is a tree. These trees will be used as a basis to construct a cycle cover. After obtaining the cycle cover, we will split some cycles to get exactly m cycles.

Algorithm 1 describes CycleSplit in detail. CycleSplit will choose the best of all feasible cycle covers in algorithm.

Algorithm 1 CycleSplit

```
input : $G = (V, E)$, $d : E \rightarrow \mathbb{R}^+$, $m$ sensors
output: A cycle cover $\mathscr{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$
1  $E_0 \leftarrow \emptyset$; $G_0 \leftarrow (V, E_0)$; $i \leftarrow 0$;
2  foreach $e \in E$ (chosen in ascending order by $d(\cdot)$) do
3      if adding $e$ to $G_i$ does not produce a cycle then
4          $i \leftarrow i + 1$; $e_i \leftarrow e$;
5          $E_i \leftarrow E_{i-1} \cup \{e\}$; $G_i \leftarrow (V, E_i)$;
6          if # of connected components $> m$ then
7              Go to next iteration;
8          $\mathscr{C}' \leftarrow \emptyset$;
9          foreach connected component $CC$ in $G_i$ do
10             Construct a cycle $\mathcal{C}$ by Christofides' Alg. [27];
11             $\mathscr{C}' \leftarrow \mathscr{C}' \cup \{\mathcal{C}\}$;
12         Split the cycles in $\mathscr{C}'$ to get a new cycle set $\mathscr{C}^{(i)}$
           with $|\mathscr{C}^{(i)}| = m$;
13 Choose the best cycle cover from the above computed
   cycle covers, and denote it as $\mathscr{C}$;
14 return $\mathscr{C}$;
```

In Line 2-12, we continue to add edges into the edge set E_i in ascending order by length. We check if the adding edge will produce a cycle. In Line 9-11, we construct a cycle for each connected component in G_i by Christofide's algorithm. In line 12, we split some of the cycles to get a new cycle cover $\mathscr{C}^{(i)}$ with $|\mathscr{C}^{(i)}| = m$. In Line 13, we find the best answer among all $\mathscr{C}^{(i)}$.

For Line 12 in Algorithm 1, we could design an efficient splitting strategy for this step. Before this step, we have already got a cycle cover, so the task here is just determining how to split large cycle into small ones, i.e. how many sensors should be assigned to each large cycle. The solution can be achieved by a simple greedy algorithm described in Algorithm 2, whose optimality is proved in Lemma 1.

Lemma 1: Algorithm 2 can find an optimal sensor allocation for a given cycle cover $\mathscr{C}$ efficiently. Here the optimality means that $\max \frac{d(\mathcal{C}_i)}{m_i}$ *is minimized.*

Proof: Assume on the contrary there is a different sensor allocation $\mathcal{A}' = \{m_1', m_2', \ldots, m_t'\}$ such that

$$\max_{1 \le i \le t} \frac{d(\mathcal{C}_i)}{m_i'} < \max_{1 \le i \le t} \frac{d(\mathcal{C}_i)}{m_i}.$$

Since $\mathcal{A}'$ is different from $\mathcal{A}$, then there exists a k such that $m_k' < m_k$ and $\frac{d(\mathcal{C}_k)}{m_k'} \ge \frac{d(\mathcal{C}_k)}{m_k}$. In Algorithm 2, we increased the value of m_k from m_k' to $m_k' + 1$, which means that $\frac{d(\mathcal{C}_k)}{m_k'}$

Algorithm 2 GreedyAllocation

```
input : $m$ sensors, a cycle cover $\mathscr{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t\}$
        and a metric $d$
output: a sensor allocation for $\mathscr{C}$ and a cycle cover $\mathscr{C}'$
        with $|\mathscr{C}'| = m$
1 $m_i \leftarrow 1$ for $1 \le i \le t$;
2 while $m > \sum_{i=1}^{t} m_i$ do
3     $k \leftarrow \arg\max_{1 \le i \le t} \frac{d(\mathcal{C}_i)}{m_i}$; $m_k \leftarrow m_k + 1$;
4 Split each $\mathcal{C}_i$ into $m_i$ segments evenly; For each segment,
  add the edge connecting its two outmost vertices to form
  a cycle and remove the redundant part; Put these $m_i$ new
  cycles into $\mathscr{C}'$;
5 return $\mathcal{A} = \{m_1, m_2, \ldots, m_t\}$ and $\mathscr{C}'$;
```

is the maximum among $\{\frac{d(\mathcal{C}_i)}{m_i}\}$ at some iteration. Notice that the value of $\max_{1 \le i \le t} \frac{d(\mathcal{C}_i)}{m_i}$ is non-increasing during the iterations. Therefore, we have

$$\frac{d(\mathcal{C}_k)}{m_k'} \ge \max_{1 \le i \le t} \frac{d(\mathcal{C}_i)}{m_i}$$

which contradicts the assumption. □

Now we will prove the approximation ratio for the CycleSplit algorithm. Denote the optimal solution as $\mathscr{C}^* = \{\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_m^*\}$, set $\text{OPT} = \max_i(d(\mathcal{C}_i^*))$. We use d^*_{max} to denote the maximal distance between any two vertices in the same cycle, i.e. $d^*_{max} = \max_{u,v \in \mathcal{C}_i^*}\{d(u, v)\}$, where u, v belongs to the vertices of some $\mathcal{C}_i^*$. We first derive an upper bound for d^*_{max}.

Lemma 2: $d^*_{max} \le \frac{1}{2}$*OPT.*

Proof: Suppose that $d(u, v) = d^*_{max}$ for u, v in some $\mathcal{C}_i^*$. To construct a cycle, we must travel along a path from u to v, then back from v to u. Since we are considering metric space, both of the two paths are larger than or equal to $d(u, v)$. Therefore

$$\text{OPT} \ge d(\mathcal{C}_i^*) \ge 2\ d(u, v) = 2\ d^*_{max}.$$

Thus Lemma 2 holds. □

As shown in Algorithm 1, denote the edges added to the graph as $e_1, e_2, \ldots, e_{|V|-1}$, we have $d(e_1) \le d(e_2) \le \cdots \le d(e_{|V|-1})$. Let $G_i = (V, E_i)$ where $E_i = \{e_1, e_2, \ldots, e_i\}$. It is easy to get the following fact.

Lemma 3: G_i is a minimum spanning forest with $|V| - i$ connected components. For each connected component CC, the subgraph induced by CC is actually a minimum spanning tree for CC.

Set $j = \arg\max_{1 \le i \le |V|-1} d(e_i) \le d^*_{max}$. Suppose the connected components of G_j are $CC_1^{(j)}, CC_2^{(j)}, \ldots, CC_{|V|-j}^{(j)}$. We will use $\mathcal{T}_1^{(j)}, \mathcal{T}_2^{(j)}, \ldots, \mathcal{T}_{|V|-j}^{(j)}$ to denote their corresponding trees.

Denote $\mathcal{T}_i^*$ as the MST for the vertices in $\mathcal{C}_i^*$.

Lemma 4: $d(\mathcal{T}_i^*) \le (1 - \frac{1}{|V| - m + 1})$*OPT, for* $1 \le i \le m$.

Proof: Notice that the number of vertices in any $\mathcal{C}_i^*$ is smaller than or equal to $(|V| - m + 1)$. Additionally, there must be an edge e' in $\mathcal{C}_i^*$ such that

$$d(e') \geq \frac{d(\mathcal{C}_i^*)}{\#\text{ of vertices in }\mathcal{C}_i^*} \geq \frac{d(\mathcal{C}_i^*)}{|V|-m+1}$$

Delete e' from $\mathcal{C}_i^*$ we can get a spanning tree and by $d(\mathcal{C}_i^*) \leq \text{OPT}$, we have

$$\begin{aligned} d(\mathcal{T}_i^*) &\leq d(\mathcal{C}_i^*) - d(e') \leq \frac{(|V|-m)d(\mathcal{C}_i^*)}{|V|-m+1} \\ &\leq \frac{(|V|-m)\text{OPT}}{|V|-m+1} \end{aligned}$$

Thus the lemma holds. □

Lemma 5: All vertices of $\mathcal{C}_i^*$ belongs to the same connected component of G_j, for $1 \leq i \leq |V|-j$.

Proof: Suppose that $\mathcal{C}_i^*$ use an edge e' to connect two different connected components of G_j. Recall that $j = \underset{1\leq i\leq |V|-1}{\arg\max}\ d(e_i) \leq d^*_{max}$. Abstract from our algorithm, $d(e') > d^*_{max}$, which contradicts the definition of d_{max}. □

Lemma 6: Suppose $CC_i^{(j)}$ contains k_i^* cycles from the optimal solution, then

$$d(\mathcal{T}_i^{(j)}) \leq \ ((\frac{3}{2} - \frac{1}{|V|-m+1})k_i^* - \frac{1}{2})OPT$$

Proof: From Lemma 5, we can find that these k_i^* cycles from the optimal solution. We can use k_i^*-1 edges to connect them, and each edge will cost no more than $d(e_j)$. Since $\mathcal{T}_i^{(j)}$ is an MST, we have

$$\begin{aligned} d(\mathcal{T}_i^{(j)}) &\leq k_i^* d(T_i^*) + \ (k_i^*-1)d(e_j) \\ &\leq k_i^* d(T_i^*) + \ (k_i^*-1)d^*_{max} \\ &\leq k_i^* \frac{(|V|-m)\text{OPT}}{|V|-m+1} + \ (k_i^*-1)\frac{\text{OPT}}{2} \\ &\leq \left((\frac{3}{2} - \frac{1}{|V|-m+1})k_i^* - \frac{1}{2}\right)\text{OPT} \end{aligned}$$

□

Lemma 7: Denote $TSP_i^{(j)}$ as the optimal TSP cycle for $CC_i^{(j)}$. Suppose $CC_i^{(j)}$ contains k_i^* cycles from the optimal solution, then $d(TSP_i^{(j)}) \leq \ (2k_i^*-1)OPT$.

Proof: Consider the k_i^* cycles from the optimal solution. We can use k_i^*-1 edges to connect them, and duplicating these edges will result in a TSP cycle. Each edge will cost no more than $d(e_j)$. Thus we have

$$\begin{aligned} d(TSP_i^{(j)}) &\leq k_i^*\text{OPT} + 2(k_i^*-1)d(e_j) \\ &\leq k_i^*\text{OPT} + 2(k_i^*-1)d^*_{max} \\ &\leq k_i^*\text{OPT} + \ (k_i^*-1)\text{OPT} \\ &\leq \ (2k_i^*-1)\text{OPT} \end{aligned}$$

We can find that the statement holds. □

Lemma 8: Denote $\mathcal{C}_i^{\prime(j)}$ as the TSP cycle computed by Christofides' Alg. for $CC_i^{(j)}$ at Line 10 in Algorithm 1. Suppose $CC_i^{(j)}$ contains k_i^* cycles from the optimal solution, then

$$d(\mathcal{C}_i^{\prime(j)}) \leq \left((\frac{5}{2} - \frac{1}{|V|-m+1})k_i^* - 1\right)OPT$$

Proof: From Christofides' algorithm, we can get that

$$\begin{aligned} d(\mathcal{C}_i^{\prime(j)}) &\leq d(\mathcal{T}_i^{(j)}) + \frac{1}{2}d(TSP_i^{(j)}) \\ &\leq \left((\frac{3}{2} - \frac{1}{|V|-m+1})k_i^* - \frac{1}{2}\right)\text{OPT} \\ &\quad + \frac{1}{2}((2k_i^*-1)\text{OPT}) \\ &\leq \left((\frac{5}{2} - \frac{1}{|V|-m+1})k_i^* - 1\right)\text{OPT} \end{aligned}$$

This finishes the proof. □

If we split $\mathcal{C}_i^{\prime(j)}$ into k_i^* paths, then each path will have length less than $(\frac{5}{2} - \frac{1}{|V|-m+1})$OPT. Thus the cycles obtained from these paths will have length less than $(5-\frac{2}{|V|-m+1})$OPT. As we have already stated in Lemma 1, the optimal way to split these cycles can be found. Therefore, we have the following theorem.

Theorem 2: $\mathscr{C}^{(j)}$ is a $(5 - \dfrac{2}{|V|-m+1})$-approximation, i.e.

$$\max_{\mathcal{C}\in\mathscr{C}^{(j)}} d(\mathcal{C}) \leq \ (5 - \frac{2}{|V|-m+1})OPT,$$

and CycleSplit is a $(5 - \frac{2}{|V|-m+1})$-approximation for Min-Period Sweep Coverage problem.

According to Theorem 2, CycleSplit is theoretically better than the existing 5-approximation solution in [12]. However, when $|V|$ is far greater than m (i.e., the number of targets is far greater than the number of sensors), the approximation ratio of CycleSplit also becomes 5.

As the first step of CycleSplit, we have to put all the edges in ascending order. This sorting process can be done in $O(|E|log(|E|))$ steps. Assign $n = |V|$ as the number of targets in the following calculation. Considering that G is a complete graph, we have $|E| = |V| \cdot \ (|V|-1) = n(n-1)$, thus the running time of this step is $O(n^2log(n))$. According to [27], the running time of Christofides' algorithm is $O(n^3)$. In the for-loop of CycleSplit, we perform Christofides' algorithm on the scale of the entire graph for m times. Then, we perform Algorithm 2 whose average time complexity is $O(m^2)$ for m times. Therefore, the overall complexity is $O(n^2log(n) + m \cdot \ (n^3+m^2))$. Since the number of sensors m is usually smaller than the number of targets n, the time complexity of CycleSplit should be $O(mn^3)$. (Note that the last step of choosing the best cycle cover can be done in constant time if we keep record of the current best result in the process.)

B. MPSC: One Dimensional Case

In this section, we will introduce an optimal algorithm called DP-MPSC for Min-Period Sweep Coverage problem in one dimensional case. In this scenario, all targets and mobile sensors are distributed along a straight line.

Given $G = (V,E)$, we denote p_i as each point belong to V. We enumerate the $p_i \in V$ from one endpoint to another endpoint of the straight line. Recall that $d(p_i,p_j)$ denote the distance between p_i and p_j.

1) DP-MPSC (An Optimal Algorithm for MPSC in One Dimensional Case): We design an optimal dynamic programming named DP-MPSC to solve Min-Period Sweep Coverage problem in one dimensional case. We divide all points belong to V into several groups and the locations of points in each group are continuous along the straight line. According to the optimal substructure analysis, we assign a mobile sensor to each PoI group, double the largest distance between any two points of each group and get the optimal solution for Min-Period Sweep Coverage problem in one dimensional case. We present the state transition equation of dynamic programming as follows:

$$Len(i,j) = \min_{k<i} \max \ (Len(k,j-1), 2d(p_i,p_{k+1}))$$

Where j firstly loops from 1 to m, i secondly loops from 1 to $|V|$, $Len(i,j)$ denotes the optimal answer when use j mobile sensors to cover the former i PoIs $(p_1,p_2,\ldots,p_i)$.

We present the pseudo code of DP-MPSC for Min-Period Sweep Coverage in Algorithm 3. Firstly the input of DP-MPSC is $G(V,E)$ and the number of mobile sensors. All distances are ordered by their locations in Line 1. Finally we use the state transition equation to calculate the value of $Len(i,j)$ in Line 2-9.

All routes are straight lines and mobile sensors move along the line back and forth.

Algorithm 3 DP-MPSC

```
input  : $G=(V,E)$, $d: E \to \mathbb{R}^+$, $m$ sensors
output: A cycle cover A cycle cover
        $\mathscr{C}=\{\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_m\}$
1  Sort the $p_i \in G$ in order;
2  foreach $j \leftarrow 1$; $j \le m$; $j++$ do
3      foreach $i \leftarrow j$; $i \le |V|$; $i++$ do
4          $\mathscr{C}_j^i \leftarrow \emptyset$;
5          foreach $k \leftarrow j-1$; $k < i$; $k++$ do
6              if $Len(i,j) > \max(Len(k,j-1), 2d(p_i,p_{k+1}))$ then
7                  $Len(i,j) \leftarrow \max(Len(k,j-1), 2d(p_i,p_{k+1}))$;
8                  Construct a cycle $\mathcal{C}'$ from the set $\{p_i,p_{i-1},\ldots,p_{k+1}\}$;
9                  $\mathscr{C}_j^i \leftarrow \mathscr{C}_{j-1}^k \cup \{\mathcal{C}'\}$;
10 return $\mathscr{C}_m^{|V|}$;
```

Theorem 3: DP-MPSC is an optimal algorithm for Min-Period Sweep Coverage problem in one dimensional case.

Proof: It is obvious that neither of two sweep routes are overlapped and PoIs on one sweep route should be continuous. Therefore the optimal solution for Min-Period Sweep Coverage problem in one dimensional case is to partition all PoIs and mobile sensors into several groups, and the locations of PoIs in each group must be continuous, which is the goal of DP-MPSC. We prove we can compute $Len(i,j)$ correctly if we have known the optimal solution for all $Len(k,j-1)$, where $k<i$. Because the PoIs in one group are continuous, the group including p_i must be $p_{k+1},p_{k+2},\ldots,p_i$ where $k<i$. Since we have known all $Len(k,j-1)$, we do not care about the partition of the former k PoIs and $j-1$ mobile sensros. Each enumeration with k can get one solution for our problem, $\max(Len(k,j-1),2d(p_i,p_{k+1}))$. After all enumerations, we have tried all partitions and the minimum of all enumerations must be the optimal solution for $Len(i,j)$. Therefore $Len(|V|,m)$ is the optimal answer for Min-Period Sweep Coverage problem in one dimensional case. Therefore Theorem 3 holds. □

According to Theorem 3, we know that DP-MPSC can get an optimal solution for Min-Period Sweep Coverage problem in one dimensional case. DP-MPSC runs in three for-loops. Therefore the time complexity of DP-MPSC is $O(n^2m)$. With the help of some advanced data structure (Segment tree), we can reduce the time complexity to $O(nmlog(n))$.

V. Heterogeneous Velocity Min-Period Sweep Coverage

In this section we will introduce a variant of Min-Period Sweep Coverage problem. Considering that the velocities of mobile sensors are not homogenous in practical scenario, we take the heterogeneous velocities into account. To make the problem more general, we change the problem into another version and normalize it.

A. Problem Formulation

In this problem, we need a new parameter $\mathscr{W} = \{\mathcal{W}_1,\mathcal{W}_2,\ldots,\mathcal{W}_m\}$. The basic idea of our algorithms to solve the Heterogeneous Velocity Min-Period Sweep Coverage problem follows the following procedures:

1) Construct a cycle cover $\mathscr{C}$;
2) Try to allocate some mobile wireless sensors to each cycle in $\mathscr{C}$ and find the optimal distribution.
3) Split the cycle according the velocities of mobile sensors and construct new cycle for each mobile wireless sensor.

Definition 7 (Heterogeneous Velocity Min-Period Sweep Coverage (HVMPSC)): Given input quadruple $(G,m,d,\mathscr{W})$, where $G=(V,E)$ is a complete graph, m is the number of sensors, $d: V\times V \to \mathbb{R}^+$ is a metric, $\mathscr{W}$ is a weight function that is inversely proportional to the velocities of mobile sensors and $\mathcal{W}_1 < \mathcal{W}_2 < \cdots < \mathcal{W}_m = 1$. Heterogeneous Velocity Min-Period Sweep Coverage (HVMPSC) problem aims to find a cycle cover $\mathscr{C}$ such that $|\mathscr{C}|=m$ and $\max_{1\le i\le|\mathscr{C}|} \mathcal{W}_i d(\mathcal{C}_i)$ is minimized.

Besides, to be more practical, we assume that $\mathcal{W}_1 > \frac{1}{\alpha}$.

B. HeteroCycleSplit: An Approximation for HVMPSC

In this section, we will introduce a approximation HeteroCycleSplit to solve Heterogeneous Velocity Min-Period Sweep Coverage problem. The basic idea of HeteroCycleSplit is shown as follows:

1) Add edges into graph to construct connected components.
2) Construct cycles for these connected components.

3) Find the optimal sensor allocation for these connected components.
4) Divide the cycle for mobile sensors according to $\mathcal{W}_i$.
5) Connect two endpoints of each route to construct a new cycle for each mobile sensor.

We denote $\mathscr{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots\}$ as a sensor allocation for each cycle. We denote $\mathcal{A}_i = \{\mathcal{W}_1^i, \mathcal{W}_2^i, \dots\}$ and $\sum_i |\mathcal{A}_i| = m$.

The detail of HeteroCycleSplit is shown in Algorithm 4. In Line 3-7, we add edges into set in ascending order by the weight of edges. In Line 5, we can get G_i, which consists of several connected components. In Line 8-11, we construct TSP cycle for each connected component. In Line 12, we find the optimal sensor allocation for the cycle cover. In Line 13, we calculate the output for the cycle cover and sensor allocation. In Line 14-15, we output the solution for HeteroCycleSplit.

Algorithm 4 HeteroCycleSplit

```
   input  : $G = (V, E)$, $d : E \to \mathbb{R}^+$, $m$ sensors, $\mathscr{A}$
   output: A cycle cover $\mathscr{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$, a sensor
           allocation $\mathscr{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m\}$
1  $E_0 \leftarrow \emptyset$; $G_0 \leftarrow (V, E_0)$; $i \leftarrow 0$;
2  foreach $e \in E$ (chosen in ascending order by $d(\cdot)$) do
3      if adding $e$ to $G_i$ does not produce a cycle then
4          $i \leftarrow i + 1$; $e_i \leftarrow e$;
5          $E_i \leftarrow E_{i-1} \cup \{e\}$; $G_i \leftarrow (V, E_i)$;
6          if # of connected components $> m$ then
7              Go to next iteration;
8          $\mathscr{C}' \leftarrow \emptyset$;
9          foreach connected component $CC$ in $G_i$ do
10             Construct a cycle $\mathcal{C}$ by Christofides' Alg. [27];
11             $\mathscr{C}' \leftarrow \mathscr{C}' \cup \{\mathcal{C}\}$;
12         Find the optimal sensor allocation $\mathscr{A}'$ for $\mathscr{C}'$ such
           that $\max_{1 \le i \le |\mathscr{C}'|} \frac{d(\mathcal{C}_i')}{\sum_{j=1}^{|\mathcal{A}_i'|} \frac{1}{\mathcal{W}_j^i}}$ is minimized;
13         Split the cycles in $\mathscr{C}'$ to get a new cycle set $\mathscr{C}^{(i)}$
           with $|\mathscr{C}^{(i)}| = m$ and allocate the corresponding
           $\mathcal{W}^{(i)}$ to $\mathscr{A}^{(i)}$;
14 Choose the best cycle cover from the above computed
   cycle covers, and denote it and the corresponding sensor
   allocation as $\mathscr{C}$ and $\mathscr{A}$;
15 return $\mathscr{C}$ and $\mathscr{A}$;
```

In order to justify the objective for sensor allocation in Line 12, we need to prove the following lemma.

Lemma 9: Given a cycle $\mathcal{C}_i$ and its sensor allocation $\mathcal{A}_i = \{\mathcal{W}_1^i, \mathcal{W}_2^i, \dots, \mathcal{W}_{t_i}^i\}$, if we split $\mathcal{C}_i$ into $|\mathcal{A}_i| = t_i$ paths (i.e., $\{\mathcal{P}_1^{(i)}, \mathcal{P}_2^{(i)}, \dots, \mathcal{P}_{t_i}^{(i)}\}$), and make sure that $\mathcal{W}_j^i d(\mathcal{P}_j^{(i)}) = \mathcal{W}_k^i d(\mathcal{P}_k^{(i)})$, for $1 \le j < k \le t_i$. Then

$$\frac{d(\mathcal{C}_i)}{\sum_{j=1}^{t_i} \frac{1}{\mathcal{W}_j^i}} = \mathcal{W}_k^i d(\mathcal{P}_k^{(i)}),$$

for $1 \le k \le t_i$.

Proof: Since we get $\{\mathcal{P}_1^{(i)}, \mathcal{P}_2^{(i)}, \dots, \mathcal{P}_{t_i}^{(i)}\}$ from the cycle $\mathcal{C}_i$,

$$\sum_{j=1}^{t_i} d(\mathcal{P}^{(i)}) = d(\mathcal{C}_i)$$

Suppose $\mathcal{W}_j^i d(\mathcal{P}_j^{(i)}) = \mathcal{W}_k^i d(\mathcal{P}_k^{(i)}) = \text{CONST}$, for $1 \le j < k \le t_i$, then

$$\frac{d(\mathcal{C}_i)}{\sum_{j=1}^{t_i} \frac{1}{\mathcal{W}_j^i}} = \frac{\sum_{j=1}^{t_i} d(\mathcal{P}^{(i)})}{\sum_{j=1}^{t_i} \frac{1}{\mathcal{W}_j^i}} = \frac{\sum_{j=1}^{t_i} \frac{\text{CONST}}{\mathcal{W}_j^i}}{\sum_{j=1}^{t_i} \frac{1}{\mathcal{W}_j^i}} = \text{CONST}$$

Therefore, Lemma 9 holds. □

From the above lemma, it is clear that if we can minimize $\max_{1 \le i \le |\mathscr{C}'|} \frac{d(\mathcal{C}_i')}{\sum_{j=1}^{|\mathcal{A}_i'|} \frac{1}{\mathcal{W}_j^i}}$ in Line 12, then after splitting the cycles to get a new cycle set $\mathscr{C}^{(i)}$ in Line 13, $\max_{1 \le i \le |\mathscr{C}^{(i)}|} \mathcal{W}_i d(\mathcal{C}_i)$ will also be minimized.

HeteroCycleSplit is a 5α-approximation for Heterogeneous Velocity Min-Period Sweep Coverage problem. To prove our algorithm's correctness, we should define some variables first.

Similarly, We denote $\mathscr{C}^* = \{\mathcal{C}_1^*, \mathcal{C}_2^*, \dots, \mathcal{C}_m^*\}$ and $\mathscr{W}^* = \{\mathcal{W}_1^*, \mathcal{W}_2^*, \dots, \mathcal{W}_m^*\}$ as the optimal solution. Here we set $\text{OPT} = \max_i (\mathcal{W}_i^* d(\mathcal{C}_i^*))$. We use d_{max}^* to denote the maximal distance between any two vertices in the same cycle, i.e., $d_{max}^* = \max_{u,v \in \mathcal{C}_i^*} \{d(u,v)\}$, where u, v belongs to the vertices of some C_i^*.

Lemma 10: $d_{max}^* \le \frac{\text{OPT}}{2\mathcal{W}_1}$.

Proof: Suppose that $d(u,v) = d_{max}^*$ for u, v in some $\mathcal{C}_i^*$. To construct a cycle containing both u and v, we must travel along a path from u to v, then back from v to u. Since we are considering metric space, both of the paths are larger than or equal to $d(u,v)$. Also, by definition, we have $\mathcal{W}_i^* \ge \mathcal{W}_1$. Therefore,

$$\frac{\text{OPT}}{\mathcal{W}_1} \ge \frac{\mathcal{W}_i^* d(\mathcal{C}_i^*)}{\mathcal{W}_1} \ge d(\mathcal{C}_i^*) \ge 2d(u,v) = 2d_{max}^*.$$

Thus Lemma 10 holds. □

As shown in Algorithm 4, denote the edges added to the graph as $e_1, e_2, \dots, e_{|V|-1}$, we have $d(e_1) \le d(e_2) \le \dots \le d(e_{|V|-1})$. Let $G_i = (V, E_i)$ where $E_i = \{e_1, e_2, \dots, e_i\}$.

It is obvious that G_i is a minimum spanning forest with $|V| - i$ connected components. For each connected component CC, the subgraph induced by CC is actually a minimum spanning tree for all the vertices in CC.

Set $j = \arg\max_{1 \le i \le |V|-1} d(e_i) \le d_{max}^*$. Suppose the connected components of G_j are $CC_1^{(j)}, CC_2^{(j)}, \dots, CC_{|V|-j}^{(j)}$. We will use $\mathcal{T}_1^{(j)}, \mathcal{T}_2^{(j)}, \dots, \mathcal{T}_{|V|-j}^{(j)}$ to denote their corresponding minimal spanning trees.

Denote $\mathcal{T}_i^*$ as the MST for the vertices in $\mathcal{C}_i^*$. From Lemma 10, we can easily get the following fact.

Lemma 11: $d(\mathcal{T}_i^*) \le d(\mathcal{C}_i^*) \le \frac{\text{OPT}}{\mathcal{W}_1}$, for $1 \le i \le m$.

Lemma 12: All vertices of $\mathcal{C}_i^*$ belong to the same connected component of G_j, for $1 \le i \le |V| - j$.

Proof: Suppose that $\mathcal{C}_i^*$ uses an edge e' to connect two different connected components of G_j. Recall that

$j = \arg\max_{1\le i\le |V|-1} d(e_i) \le d^*_{max}$. This will lead to $d(e') > d^*_{max}$, which contradicts the definition of d^*_{max}. Therefore, Lemma 12 holds. □

Lemma 13: Suppose $CC_i^{(j)}$ contains t_i^* cycles from the optimal solution, then

$$d(\mathcal{T}_i^{(j)}) \le (\frac{3}{2}t_i^* - \frac{1}{2})\frac{\text{OPT}}{\mathcal{W}_1}.$$

Proof: From Lemma 12, we can find these t_i^* cycles from the optimal solution. Use $t^* - 1$ edges no larger than $d(e_j)$ to connect them and delete on edge from each cycle. After this method, we can get a new spanning tree. Since $\mathcal{T}_i^{(j)}$ is an MST, we have

$$\begin{aligned} d(\mathcal{T}_i^{(j)}) &\le t_i^* d(\mathcal{T}_i^*) + (t_i^*-1)d^*_{max} \\ &\le t_i^* d(\mathcal{C}_i^*) + (t_i^*-1)d^*_{max} \\ &\le t_i^* \frac{\text{OPT}}{\mathcal{W}_1} + (t_i^*-1)\frac{\text{OPT}}{2\mathcal{W}_1} \\ &\le (\frac{3}{2}t_i^* - \frac{1}{2})\frac{\text{OPT}}{\mathcal{W}_1} \end{aligned}$$

□

Lemma 14: Denote $TSP_i^{(j)}$ as the optimal TSP cycle for $CC_i^{(j)}$. Suppose $CC_i^{(j)}$ contains t_i^* cycles from the optimal solution, then

$$d(TSP_i^{(j)}) \le (2t_i^* - 1)\frac{\text{OPT}}{\mathcal{W}_1}$$

Proof: Use $t^* - 1$ edges no larger than $d(e_j)$ to connect the t_i^* cycles, and duplicate these edges will result in a TSP cycle. Each edge in the TSP cycle will cost no more than $d(e_j)$. Thus we have

$$\begin{aligned} d(TSP_i^{(j)}) &\le t_i^* d(\mathcal{C}_i^*) + 2(t_i^*-1)d(e_j) \\ &\le t_i^* \frac{\text{OPT}}{\mathcal{W}_1} + 2(t_i^*-1)\frac{\text{OPT}}{2\mathcal{W}_1} \\ &\le (2t_i^* - 1)\frac{\text{OPT}}{\mathcal{W}_1} \end{aligned}$$

□

Lemma 15: Denote $\mathcal{C}_i^{(j)}$ as the TSP cycle computed by Christofides' algorithm for $CC_i^{(j)}$ at Line 10. Suppose $CC_i^{(j)}$ contains t_i^* cycles from the optimal solution, then

$$d(\mathcal{C}_i^{(j)}) \le (\frac{5}{2}t_i^* - 1)\frac{\text{OPT}}{\mathcal{W}_1}.$$

Proof: From Christofides' algorithm, we can get that

$$\begin{aligned} d(\mathcal{C}_i^{(j)}) &\le d(\mathcal{T}_i^{(j)}) + \frac{1}{2}d(TSP_i^{(j)}) \\ &\le (\frac{3}{2}t_i^* - \frac{1}{2})\frac{\text{OPT}}{\mathcal{W}_1} + \frac{1}{2}(2t_i^*-1)\frac{\text{OPT}}{\mathcal{W}_1} \\ &\le (\frac{5}{2}t_i^* - 1)\frac{\text{OPT}}{\mathcal{W}_1} \end{aligned}$$

This finished the proof. □

From the above lemma, since $\frac{1}{\alpha} < \mathcal{W}_1 \le \mathcal{W}_k^i \le \mathcal{W}_m = 1$, we can get that

$$\begin{aligned} \frac{d(\mathcal{C}_i^{(j)})}{\sum_{k=1}^{t_i^*}\frac{1}{\mathcal{W}_k^i}} &\le (\frac{5}{2}t_i^* - 1)\frac{\text{OPT}}{\mathcal{W}_1\sum_{k=1}^{t_i^*}\frac{1}{\mathcal{W}_k^i}} \\ &\le (\frac{5}{2}t_i^* - 1)\frac{\text{OPT}}{\mathcal{W}_1\frac{t_i^*}{\mathcal{W}_m}} \\ &\le (\frac{5}{2}\alpha - \frac{\alpha}{t_i^*})\text{OPT} \end{aligned}$$

If we split $\mathcal{C}_i^{(j)}$ into t_i^* paths and assign a weight to each path, satisfying the conditions in Lemma 9, then after being multiplied by its own weight, each path will give a value less than $\frac{5}{2}\alpha\cdot\text{OPT}$. Thus, the cycles obtained from these paths will each give a value less than $5\alpha\cdot\text{OPT}$. Therefore, we have the following theorem.

Theorem 4: $\mathscr{C}^{(j)}$ is a 5α-approximation, i.e.,

$$\max_{\mathcal{C}\in\mathscr{C}^{(j)}} \mathcal{W}_{\mathcal{C}} d(\mathcal{C}) \le 5\alpha\cdot\text{OPT},$$

and HeteroCycleSplit is a 5α-approximation for Heterogeneous Velocity Min-Period Sweep Coverage problem.

Similar to CycleSplit, the overall time complexity of HeteroCycleSplit is also $O(mn^3)$ for simplicity.

VI. Connected Path Min-Period Sweep Coverage

In this section, we propose a new variant of Min-Period Sweep Coverage. We formally give a definition for Connected Path Min-Period Sweep Coverage (CPMPSC) problem and propose a 2-approximation called PathSplit for this problem.

A. Problem Formulation

In this part, we will give a formal definition for Connected Path Min-Period Sweep Coverage problem.

Definition 8 (Connected Path Min-Period Sweep Coverage (CPMPSC)) Given input triple (G, m, d), where $G = (V, E)$ is an undirected connected graph, m is the number of sensors, $d : V \times V \to \mathbb{R}^+$ is a metric, Connected Path Min-Period Sweep Coverage (CPMPSC) problem aims to find a path cover $\mathscr{P}$ such that $\bigcup \mathcal{P}_i = E$, $|\mathscr{P}| = m$ and $\max_{1\le i\le|\mathscr{P}|} d(\mathcal{P}_i)$ is minimized.

Besides, we must define the Chinese Postman Problem [43].

Definition 9 (Chinese Postman Problem): Given $G(V, E)$, double some edges $e \in E$ to make each point $v \in V$ has even degree or just two points have odd degree. The total length of all these edges is minimized.

We can use the algorithm in [43] to get optimal solution for Chinese Postman Problem. The algorithm can be described as follows.

1) First determine for every pair $v_i, v_j \in V$ of odd nodes the *shortest path* P_{ij} joining these two nodes and define d_{ij} to be the length of path P_{ij}.
2) Construct the complele graph $\overline{G} = (\overline{V}, \overline{E})$ where $\overline{V}$ denotes the set of all odd degree nodes in G. Associate with every edge e_{ij} joining $v_i, v_j \in \overline{V}$ the edge weight d_{ij} and solve the associated *stable marriage problem* (SMP).

3) The edges e_{ij} of the optimal matching correspond to the path P_{ij} the edges of which have to be duplicated to obtain the optimal postman tours.
4) Finally, construct an Eulerian path in G as the optimal solution to the problem.

B. PathSplit: An Approximation for CPMPSC

In this section, we will introduce a 2-approximation called PathSplit for Connected Path Min-Period Sweep Coverage problem. The basic idea for this problem is shown as follow:

1) Construct a Chinese Postman Path for the whole graph $G(V, E)$
2) Divide the path into m paths and assign each path a mobile wireless sensor.

The algorithm is shown as follows in Algorithm 5.

Algorithm 5 PathSplit

```
input  : $G = (V, E)$, $d : E \to \mathbb{R}^+$, $m$ sensors
output: A Path cover $\mathscr{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m\}$
1  $\mathcal{P}' \leftarrow$ an optimal path for the Chinese Postman
   Problem instance on $G$;
2  $Len \leftarrow \frac{d(\mathcal{P}')}{m}$;
3  $\mathscr{P} \leftarrow \emptyset$;
4  while $d(\mathcal{P}') > Len$ do
5      $\mathcal{P}" \leftarrow \emptyset$;
6      foreach vertex $v \in \mathcal{P}'$ in order along the path $P$ do
7          if $d(\mathcal{P}") < Len$ then
8              Add $v$ into $\mathcal{P}"$;
9      $\mathcal{P}' = \mathcal{P}' \backslash \mathcal{P}"$;
10     Add $\mathcal{P}"$ into $\mathscr{P}$;
11 return $\mathscr{P}$;
```

In Line 1, we produce an optimal path for the Chinese Postman Problem instance on G. In Line 4-10, we divide the optimal path for Chinese Postman Problem to construct m paths for Connected Path Min-Period Sweep Coverage problem, which is restricted by Len computed from the length of the optimal path for the Chinese Postman Problem.

We denote the optimal solution as $\mathscr{P}^* = \{\mathcal{P}_1^*, \mathcal{P}_2^*, \ldots, \mathcal{P}_m^*\}$, and set OPT $= \max\limits_{1 \le i \le |\mathscr{P}^*|} d(\mathcal{P}_i^*)$. Then we have the following theorem.

Theorem 5: Algorithm *PathSplit* is a 2-approximation algorithm.

Proof: The optimal solution uses m paths and totally covers E. Therefore, $d(E) \le m$OPT. To construct $\mathcal{P}'$ from G, we double some of the edges in E, therefore, $d(\mathcal{P}') \le 2d(E)$, and our output is $\frac{d(\mathcal{P}')}{m}$. We can get

$$\frac{d(E)}{m} \le \text{OPT} \le \frac{d(\mathcal{P}')}{m} \le \frac{2d(E)}{m} \le 2\text{OPT}.$$

Therefore theorem 5 holds. □

According to the algorithm in [43], the optimal path for the Chinese Postman Problem can be calculated in $O(n^3)$ computational steps. Besides, the path division takes $O(n)$ time to complete. Therefore, the growth rate of PathSplit is $O(n^3)$.

VII. Performance Evaluation

In this section, we evaluate and compare the performance of our algorithms and other algorithms on a stand alone C++ simulation platform. We prepare a 200×200 virtual 2-D place for simulation and randomly deploy a number of targets.

A. Performance Evaluation of CycleSplit

In this section, we evaluate the performance of CycleSplit. We compare our algorithms with OSweep in [15], MinExpand proposed in [15], and PDBA in [16]. In this simulation, the number of targets range from 20 to 500 with a step of 20 and the number of mobile sensors is $\frac{1}{20}$, $\frac{1}{10}$, $\frac{3}{20}$ or $\frac{1}{5}$ of the number of targets. When the numbers of mobile sensors and targets are fixed, we generate 20 problem instances and calculate the average of the costs of the outputs of CycleSplit, OSweep [15], MinExpand [15], and PDBA [16].

Since some previous algorithms are to find the minimum number of mobile sensors to achieve Sweep Coverage with a constraint on the length of mobile sensors' trajectories [15], [16], we use binary search through changing the bound length for sweep coverage until the output of the previous algorithms equals to the number of mobile sensors we input. We use that bound length as the output of these algorithms, which is shown in Algorithm 6.

Algorithm 6 Simulation

```
input  : $G = (V, E)$, $d : E \to \mathbb{R}^+$, $m$ sensors
output: $Len$
1 $L_l \leftarrow 0$;
2 $L_r \leftarrow d(E)$;
3 while $l + \varepsilon < r$ do
4     $M \leftarrow \frac{(L_l + L_r)}{2}$;
5     if $TestAlgorithm(M) \le m$ then
6         $L_r \leftarrow M$;
7     else
8         $L_l \leftarrow M$;
9 return $M$;
```

In Line 1-2, the algorithm initialize the left and the right bounds for the output. In Line 3-8, the algorithm use binary search to find the optimal length under the constraint of which the output is equal to the number of given mobile wireless sensors. The output is M in Line 9.

Figure 2 (a), Figure 2 (b), Figure 2 (c) and Figure 2 (d) are simulation results for these four algorithms under different parameter settings. The x axis denotes the number of targets to be covered, while the y axis shows the length of the longest trajectory calculated by each algorithm. We can easily abstract from these four figures that CycleSplit algorithm outperforms previous algorithms.

We can also find that when the number of targets become larger, the difference of the largest trajectory length among four algorithms become smaller. The reason for why this phenomenon occurs is that given the virtual area, a large number of targets will result in the trajectory for each mobile sensor becoming small. It is the same with the difference among four

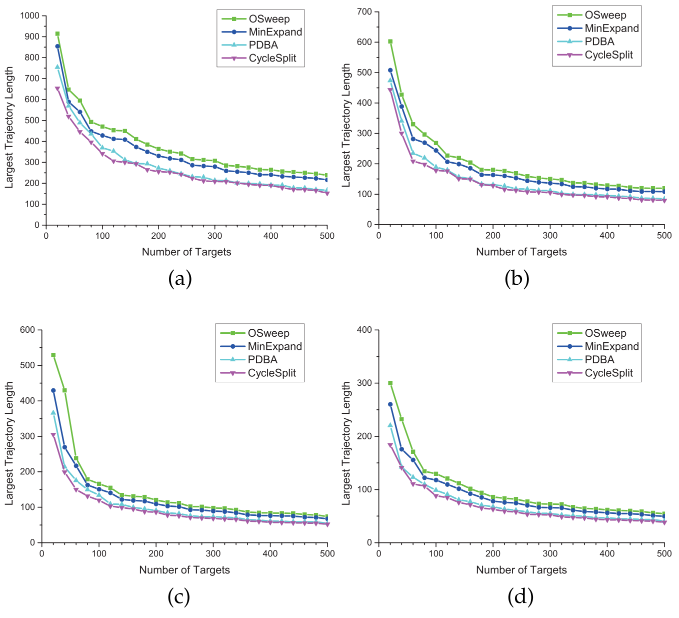

Fig. 2. Simulation results of four algorithms. (a) $m = \frac{n}{20}$. (b) $m = \frac{n}{10}$. (c) $m = \frac{3n}{20}$. (d) $m = \frac{n}{5}$.

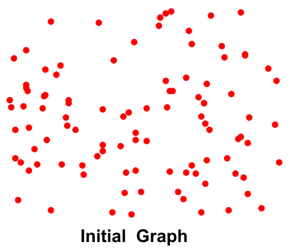

Fig. 3. 100 Targets to be covered.

algorithms. Besides, the TSP length for each sensor can be more accurate when more targets and more mobile sensors are distributed in the given area. When fixing the ratio between the number of mobile sensors and targets, the largest trajectory length becomes smaller with the increment of the number of targets.

Figure 3 exhibits an instance with $n = 100$ targets and we want to cover them with $m = 20$ sensors. Figure 4 (a)-(d) show the output trajectories computed by these four algorithms. We can find that the output of CycleSplit has the shortest total length, which also definitely proves that the performance of our algorithm is better than the performance of former algorithms.

B. Performance Evaluation of HeteroCycleSplit

In this section, we evaluate the performance of HeteroCycleSplit and compare our algorithms with k-ITSPN proposed in [14]. In this simulation, the number of targets ranges from 40 to 200 with a step of 20 and the number of mobile sensors are 5, 10, 15 and 20. When the numbers of mobile sensors and targets are fixed, we generate 20 problem instances and calculate the average of the costs of the outputs of HeteroCycleSplit, k-ITSPN [14]. The weight of each mobile sensor for $m = 5$ are from 0.6 to 1 with a step of 0.1. The weight of each mobile sensor for $m = 10$ are from 0.55 to 1 with a step of 0.05. The weight of each mobile sensor for

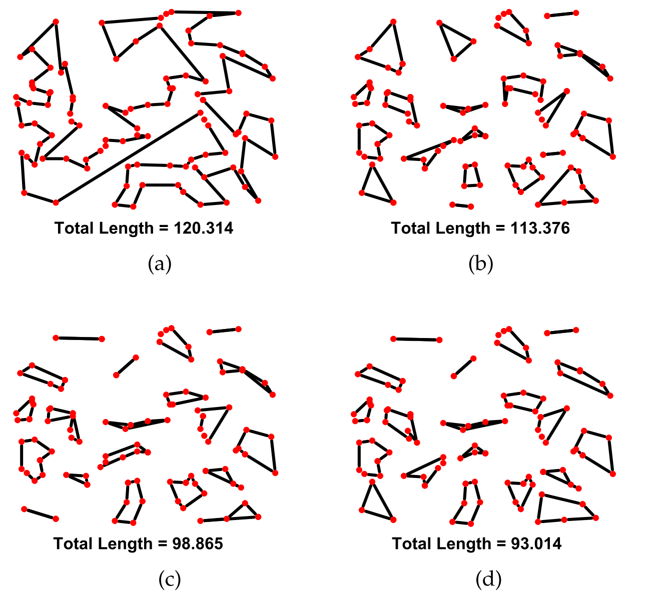

Fig. 4. Outputs for one instance of MPSC. (a) Result by OSweep. (b) Result by MinExpand. (c) Result by PDBA. (d) Result by CycleSplit.

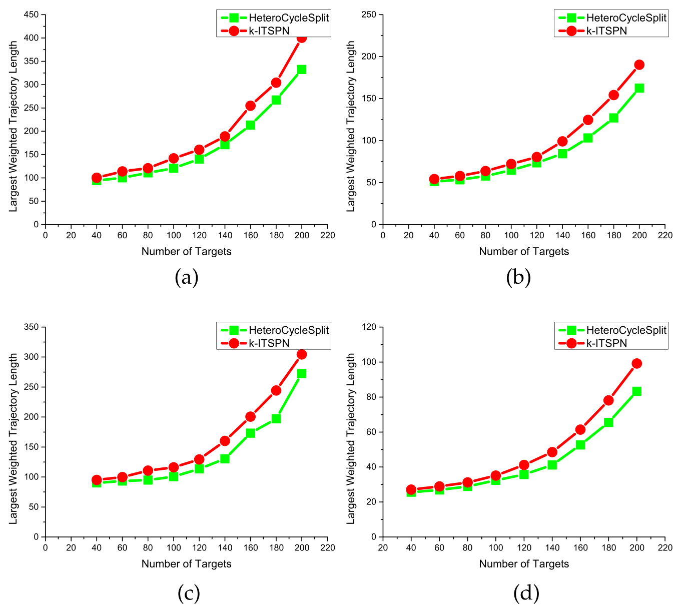

Fig. 5. Simulation results of HeteroCycleSplit and k-ITSPN. (a) $m = 5$. (b) $m = 10$. (c) $m = 15$. (d) $m = 20$.

$m = 15$ are from 0.533 to 1 with a step of 0.033. The weight of each mobile sensor for $m = 20$ are from 0.525 to 1 with a step of 0.025

The output data can be seen in Figure 5 (a), Figure 5 (b), Figure 5 (c) and Figure 5 (d). Abstract from Figure 5, we can find that our algorithm, HeteroCycleSplit is better than k-ITSPN. When fixing the number of mobile sensors, the largest weighted trajectory length becomes larger with the increment of the number of targets.

The instance in Figure 3 is employed again and we want to cover them with $m = 5$ sensors. The weight of each mobile sensor is $\{0.6, 0.7, 0.8, 0.9, 1.0\}$. Figure 6 (a)-(b) show the output trajectories computed by these two algorithms. The targets of input data is the same as those in the former section.

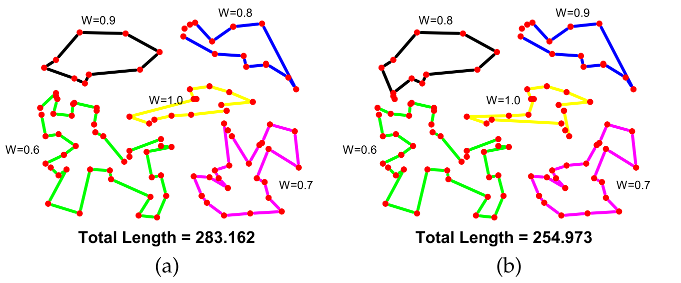

Fig. 6. Outputs for one instance of HVMPSC. (a) Result by k-ITSPN. (b) Result by HeteroCycleSplit.

The figure shows that the output of k-ITSPN is 283.162. Meanwhile, the output of HeteroCycleSplit is 254.973. We can find that the output of HeteroCycleSplit has the shorter total weighted length.

VIII. Conclusion and Future Work

In this paper, we discuss three variations of the sweep coverage problem to minimize the longest sweep period of mobile sensors, and design three constant-factor approximations respectively.

For min-period sweep coverage problem (MPSC), we propose a $(5 - \frac{2}{n-m+1})$-approximation named CycleSplit, which improves the the best known approximation ratio of 5. For heterogeneous velocity min-period sweep coverage problem (HVMPSC), we propose a 5α-approximation named HeteroCycleSplit. For connected path min-period sweep coverage problem (CPMPSC), we propose a 2-approximation named PathSplit. There has been no algorithm proposed for the last problem mentioned above until now. Former approximation algorithm in [14] deals with the k-ITSPN problem which is similar to the HVMPSC problem. However, it cannot be used in all scenarios. Our algorithm may not be so efficient, but it does not have such limitation. We also propose an optimal algorithm DP-MPSC for Min-Period Sweep Coverage problem in one dimensional case. It is solved by dynamic programming. With the help of Segment Tree, we can largely improve the efficiency of our optimal algorithms. Our analysis is based on the property of metric space, which is more general than the commonly discussed Euclidean space.

Finally, we compare our algorithms with several previous works by simulations. We compare our algorithm CycleSplit with OSweep, PDBA and MinExpand. We also compare our algorithm HeteroCycleSplit with k-ITSPN. Both theoretical analysis and numerical experiments validate the efficiency of our design. We also give some examples for some given instances of these problems.

In future, we will discuss the problems in more practical scenarios. We may take even more factors into consideration, such as data capacity and energy consumption of mobile sensors. Besides, we may also consider efficient data collecting or data gathering mechanisms under these circumstances.

Acknowledgement

The authors would like to thank X. Zhu and Y. Feng for their contribution on the early versions of this paper.

References

[1] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Mar. 2005, pp. 1976–1984.

[2] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MOBICOM)*, 2005, pp. 284–298.

[3] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MOBIHOC)*, 2008, pp. 411–420.

[4] A. Chen, S. Kumar, and T. H. Lai, "Designing localized algorithms for barrier coverage," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MOBICOM)*, 2007, pp. 63–74.

[5] M. Amac Guvensan and A. Gokhan Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1238–1255, 2011.

[6] Y. Wang and G. Cao, "On full-view coverage in camera sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2011, pp. 1781–1789.

[7] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Cost-effective barrier coverage by mobile sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Mar. 2012, pp. 819–827.

[8] 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.

[9] B. Gorain and P. S. Mandal, "Approximation algorithms for sweep coverage in wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 74, no. 8, pp. 2699–2707, 2014.

[10] M. Li, W. Cheng, K. Liu, Y. He, X. Li, and X. Liao, "Sweep coverage with mobile sensors," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1534–1545, Nov. 2011.

[11] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic event capture using mobile sensors subject to a quality metric," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MOBICOM)*, 2006, pp. 98–109.

[12] W. Yu and Z. Liu, "Improved approximation algorithms for min-max and minimum vehicle routing problems," in *Proc. Int. Comput. Combinat. Conf. (COCOON)*, 2015, pp. 147–158.

[13] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," *SIAM J. Comput.*, vol. 7, no. 2, pp. 178–193, 1978.

[14] L. Xue *et al.*, "Multiple heterogeneous data ferry trajectory planning in wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 2274–2282.

[15] J. Du, Y. Li, H. Liu, and K. Sha, "On sweep coverage with minimum mobile sensors," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2010, pp. 283–290.

[16] B. H. Liu, N. T. Nguyen, and V. T. Pham, "An efficient method for sweep coverage with minimum mobile sensor," in *Proc. IEEE Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Aug. 2014, pp. 289–292.

[17] B. Gorain and P. S. Mandal, "Approximation algorithm for sweep coverage on graph," *Inf. Process. Lett.*, vol. 115, no. 9, pp. 712–718, 2015.

[18] Y. Feng, X. Gao, F. Wu, and G. Chen, "Shorten the trajectory of mobile sensors in sweep coverage problem," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[19] S. Li, L. Z. Wang Wei, L. Feng, and Z. Jiliu, "A sweep coverage scheme based on vehicle routing problem," *Indonesian J. Elect. Eng.*, vol. 11, no. 4, pp. 2029–2036, 2013.

[20] R. Moazzez-Estanjini and I. C. Paschalidis, "On delay-minimized data harvesting with mobile elements in wireless sensor networks," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1191–1203, 2012.

[21] D. Zhao, H. Ma, and L. Liu, "Mobile sensor scheduling for timely sweep coverage," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 1771–1776.

[22] Z. Chen *et al.*, "Efficient scheduling strategies for mobile sensors in sweep coverage problem," in *Proc. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2016, pp. 1–4.

[23] B. Gorain and P. S. Mandal, "Line sweep coverage in wireless sensor networks," in *Proc. Commun. Syst. Netw. (COMSNETS)*, Jan. 2014, pp. 1–6.

[24] B. Gorain and P. S. Mandal, "Point and area sweep coverage in wireless sensor networks," in *Proc. IEEE Int. Symp. Modeling, Optim. Mobile, Ad Hoc Wireless Netw. (WiOpt)*, May 2013, pp. 140–145.

[25] C. Liu, H. Du, and Q. Ye, "Sweep coverage with return time constraint," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[26] P. Huang, F. Lin, C. Liu, J. Gao, and J. Zhou, "ACO-based sweep coverage scheme in wireless sensor networks," *J. Sensors*, vol. 2015, 2015, Art. no. 484902. [Online]. Available: https://www.hindawi.com/journals/js/2015/484902/

[27] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Graduate School Ind. Admin., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. 388, 1976.

[28] M. Karpinski, M. Lampis, and R. Schmied, "New inapproximability bounds for TSP," *J. Comput. Syst. Sci.*, vol. 81, no. 8, pp. 1665–1677, 2015.

[29] G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha, "Min–max tree covers of graphs," *Oper. Res. Lett.*, vol. 32, no. 4, pp. 309–315, 2004.

[30] M. R. Khani and M. R. Salavatipour, "Improved approximation algorithms for the min-max tree cover and bounded tree cover problems," *Algorithmica*, vol. 69, no. 2, pp. 443–460, 2014.

[31] Z. Xu and Q. Wen, "Approximation hardness of min–max tree covers," *Oper. Res. Lett.*, vol. 38, no. 3, pp. 169–173, 2010.

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

[33] L. Shu, K.-W. Cheng, X.-W. Zhang, and J.-L. Zhou, "Periodic sweep coverage scheme based on periodic vehicle routing problem," *J. Netw.*, vol. 9, no. 3, pp. 726–732, 2014.

[34] S. O. Tiwari and S. K. Yadav, "Data harvesting with mobile elements in wireless sensor networks," *J. Electron. Commun. Eng.*, vol. 9, no. 1, pp. 104–114, 2014.

[35] W. Cheng *et al.*, "Sweep coverage with mobile sensors," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, Apr. 2008, pp. 1–9.

[36] 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.

[37] Z. Chen *et al.*, "A route scheduling algorithm for the sweep coverage problem," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun./Jul. 2015, pp. 750–751.

[38] T. M. Cheng, A. V. Savkin, and F. Javed, "Decentralized control of a group of mobile robots for deployment in sweep coverage," *Robot. Autom. Syst.*, vol. 59, nos. 7–8, pp. 497–507, 2011.

[39] X. Gao, X. Zhu, Y. Feng, F. Wu, and G. Chen, "Data ferry trajectory planning for sweep coverage problem with multiple mobile sensors," in *Proc. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2016, pp. 1–9.

[40] P. Huang, L. Xu, Z. Kang, C. Liu, and F. Lin, "Ga-based sweep coverage scheme in WSN," in *Proc. Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2016, pp. 254–259.

[41] C. Liu, H. Du, and Q. Ye, "Utilizing communication range to shorten the route of sweep coverage," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[42] B. Gorain and P. S. Mandal, "Solving energy issues for sweep coverage in wireless sensor networks," *Discrete Appl. Math.*, vol. 228, pp. 130–139, Sep. 2017.

[43] R. E. Burkard and U. Derigs, "The chinese postman problem," in *Assignment and Matching Problems: Solution Methods With FORTRAN-Programs*. Berlin, Germany: Springer-Verlag, 1980, pp. 72–98. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-51576-7_6

Xiaofeng Gao received the B.S. degree in information and computational science from Nankai University, China, in 2004, the M.S. degree in operations research and control theory from Tsinghua University, China, in 2006, and the Ph.D. degree in computer science from The University of Texas at Dallas, Richardson, TX, USA, in 2010. She is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. She has authored over 130 peer-reviewed papers in the related area, including well-archived international journals, such as the IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and also in well-known conference proceedings, such as SIGKDD, INFOCOM, and ICDCS. Her research interests include wireless communications, data engineering, and combinatorial optimizations. She has served on the editorial board of *Discrete Mathematics, Algorithms and Applications*, and as the PCs and peer reviewers for a number of international conferences and journals.

Jiahao Fan is an undergraduate student from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include combinatorial optimization and approximation algorithm.

Fan Wu received the B.S. degree in computer science from Nanjing University in 2004, and the Ph.D. degree in computer science and engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2009. He has visited the University of Illinois at Urbana-Champaign, Champaign, IL, USA, as a Post-Doctoral Research Associate. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He has authored over 130 peer-reviewed papers in leading technical journals and conference proceedings. His research interests include wireless networking and mobile computing, algorithmic game theory and its applications, and privacy preservation. He was a recipient of China National Natural Science Fund for Outstanding Young Scientists, CCF-Intel Young Faculty Researcher Program Award, CCF-Tencent Rhinoceros bird Open Fund, and Pujiang Scholar. He has served as the Chair of CCF YOCSEF Shanghai, on the Editorial Board of *Computer Communications*, and as the member of technical program committees of over 40 academic conferences.

Guihai Chen received the B.S. degree from Nanjing University in 1984, the M.E. degree from Southeast University in 1987, and the Ph.D. degree from the University of Hong Kong in 1997. He had been invited as a Visiting Professor by many universities, including the Kyushu Institute of Technology, Japan, in 1998, the University of Queensland, Australia, in 2000, and Wayne State University, Detroit, MI, USA, from 2001 to 2003. He is currently a Distinguished Professor with Shanghai Jiaotong University, China. He has authored over 200 peer-reviewed papers, and over 120 of them are in well-archived international journals, such as the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Journal of Parallel and Distributed Computing*, *Wireless Network*, *The Computer Journal*, *International Journal of Foundations of Computer Science*, and *Performance Evaluation*, and also in well-known conference proceedings, such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS, and ICDCS. He has a wide range of research interests with a focus on sensor network, peer-to-peer computing, and high-performance computer architecture and combinatorics.