Optimization Letters (2024) 18:575–589
https://doi.org/10.1007/s11590-023-02008-6

ORIGINAL PAPER

A unified approach to approximate partial, prize-collecting, and budgeted sweep cover problems

Wei Liang[1] · Zhao Zhang[1] · Ding-Zhu Du[2]

Received: 1 November 2022 / Accepted: 27 April 2023 / Published online: 22 May 2023

Abstract
In a sweep cover problem, mobile sensors move around to collect information from positions of interest (PoIs) periodically and timely. A PoI is *sweep-covered* if it is visited at least once in every time period t. In this paper, we study approximation algorithms on three types of sweep cover problems. The *partial sweep cover problem* (PSC) aims to use the minimum number of mobile sensors to sweep-cover at least a given number of PoIs. The *prize-collecting sweep cover problem* aims to minimize the cost of mobile sensors plus the penalties on those PoIs that are not sweep-covered. The *budgeted sweep cover problem* (BSC) aims to use a budgeted number N of mobile sensors to sweep-cover as many PoIs as possible. We propose a unified approach which can yield approximation algorithms for PSC and PCSC within approximation ratio at most 8, and a bicriteria $(4, \frac{1}{2})$-approximation algorithm for BSC (that is, no more than $4N$ mobile sensors are used to sweep-cover at least $\frac{1}{2}opt$ PoIs, where opt is the number of PoIs that can be sweep-covered by an optimal solution). Furthermore, our results for PSC and BSC can be extended to their weighted version, and our algorithm for PCSC answers a question proposed in Liang et al. (Theor Comput Sci, 2022) on PCSC.

Keywords Sweep cover · Partial cover · Prize-collecting cover · Budgeted cover · Approximation algorithm · Bicriteria approximation algorithm

✉ Zhao Zhang
hxhzz@sina.com

✉ Ding-Zhu Du
dzdu@utdallas.edu

[1] School of Mathematical Sciences, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

[2] Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA

1 Introduction

Sweep cover problems have been widely studied in Wireless Sensor Networks (WSNs). Instead of providing continuous coverage for positions of interest (PoIs), a sweep cover problem aims to visit PoIs periodically, which plays an important role in wildlife protection, environment monitoring, etc. For example, to protect wild animals, static sensors are spread in their habitat, and the information collected by the static sensors need to be gathered regularly by mobile sensors. In such a setting, mobile sensors are required to move along designed trajectories (called *sweep-routes*) to visit PoIs periodically. A PoI is sweep-covered if it is visited at least once in every time period t (t is called the *sweep-period*).

The concept of sweep cover was proposed in [14]. Cheng et al. [11] were the first to study approximation algorithm on the sweep cover problem. They proposed a *min-sensor sweep cover problem* (MinSSC), the goal of which is to use the minimum number of mobile sensors to sweep-cover *all* PoIs. They proved that MinSSC cannot be approximated within factor 2 unless $P = NP$. Nice approximation algorithms exist. For example, Gorain and Mandal [20] gave a 3-approximation algorithm for the MinSSC problem.

After that, various sweep cover models are proposed, the goals of which include minimizing the number of mobile sensors [20, 25, 31], controlling their speeds [36, 37], finding the shortest sweep-routes [9, 13, 30], minimizing the maximum sweep-period [15, 16] and maximizing the number of covered PoIs using limited mobile sensors [22, 26, 28].

Note that most previous works consider a *full* sweep cover version, requiring All PoIs to be sweep-covered. In many real applications, meeting the coverage requirement of every PoI is too expensive. It might be sufficient to sweep-cover only a part of the PoIs. For example, in the wildlife monitoring example, some distant locations may be ignored. This motivates us to propose the *partial sweep cover problem* (PSC).

Another sweep cover problem that allows some PoIs to be ignored is the *prize-collecting sweep cover problem* (PCSC) proposed by Liang et al. in [27], the goal of which is to minimize the cost of mobile sensors plus the penalty incurred by un-covered PoIs. A 5-approximation algorithm was presented assuming that there exist a constant number of base stations and every mobile sensor has to go through some base station. The reason why a constant number of base stations is assumed is because in the expression of their running time, the number of base stations appears in the exponent. In the conclusion part of [27], it was asked whether there is a constant approximation algorithm for the PCSC problem without base station assumption.

The *budgeted sweep cover problem* (BSC) also allows some PoIs to be ignored, which can be viewed as a dual version of the PSC problem. In BSC, the number of mobile sensors is given, the goal is to sweep-cover as many PoIs as possible. This problem was proposed by Huang et al. in [22]. However, they assume that the mobile sensors can only choose their routes from a given set of alternatives. They did not consider the general case. Liang et al. [28] considered the BSC problem in a special setting, when all PoIs are placed on a line. It is open whether there is a performance guaranteed approximation algorithm for the BSC problem in a general setting.

In this paper, we present a unified algorithm for PSC, PCSC and BSC, obtaining theoretically guaranteed approximation ratios.

1.1 Related work

Starting from Cheng et al. [11], sweep cover problems have been extensively studied. In [25], by a reduction from the *traveling salesman problem* (TSP), the authors proved that MinSSC cannot be approximated within factor 2 in metric graph unless $P = NP$. Assuming that the optimal sweep-route is a shortest Hamiltonian cycle and mobile sensor are uniformly deployed on the cycle to sweep-cover PoIs, they gave a 2-approximation algorithm for MinSSC. In [20], Gorain found that grouping the mobile sensors is more efficient, that is, PoIs are divided into groups, a group of PoIs are sweep-covered by a same set of mobile sensors cooperatively. It can be shown that under *grouping strategy*, the number of mobile sensors can be significantly reduced. Making use of a *minimum spanning forest* to group sensors, the authors designed a 3-approximation algorithm for MinSSC. In Gorain et al. [21] considered two types of sensors, mobile and static, with different costs. The goal is to sweep cover all PoIs using mixed sensors with the minimum total cost. By guessing the number of static sensors in an optimal solution, they gave an 8-approximation algorithm.

Considering limited energy, some researchers [10, 29] proposed *distance/ time constrained* sweep cover problems, which means that each mobile sensor can travel at most distance D (or time T) before visiting some base station for charging. Gao et al. [15] considered a dual problem of MinSSC, *min-max sweep period problem* (Min-MaxSP). Given limited mobile sensors, Min-MaxSP aims to sweep-cover all PoIs such that the maximum sweep-period is minimized. In Gao et al. [16] considered a non-cooperative version of the Min-MaxSP problem, where mobile sensors work independently along their own trajectories. In all these literatures, theoretically guaranteed approximation ratios were obtained. Note that all these works considered the full version of the sweep cover problems, that is, all PoIs are required to be sweep-covered.

The *prize-collecting sweep cover problem* (PCSC) was proposed in [27]. A 5-approximation algorithm was proposed under the assumption that there are constant number of predefined base stations and every mobile sensor has to go through some base station. The reason why the algorithm requires the number of base stations to be upper bounded by a constant is because its running time exponentially depends on the number of base stations. A question was asked in its conclusion section: is there a constant approximation algorithm for the PCSC problem without base station assumptions?

The BSC problem was first studied by Huang et al. [22], assuming that the mobile sensors can choose their routes from given alternatives. This assumption makes the problem transformed into the *budgeted set cover problem*, and thus a $(1 - \frac{1}{e})$

-approximation follows. In Nie et al. [38] studied the weighted BSC problem, the goal of which is to collect the maximum weight from sweep-covered PoIs using limited mobile sensors. Heuristic algorithms were given. When PoIs are deployed on a line, Liang et al. [28] presented the first performance guaranteed algorithms for the weighted BSC problem: if the mobile sensors have the same speed, then an optimal solution can be obtained; if the mobile sensors have a constant number of speeds, then a $\frac{1}{2}$-approximation algorithm was given; and if all sensors have different speeds, then a $\frac{1}{2}(1-\frac{1}{e})$-approximation algorithm was proposed. For the BSC problem in a general setting, we have not seen any work with theoretically guaranteed approximation ratio.

For the studies on some other variants of the sweep cover problem, the readers may refer to the survey in Chapter 11 of [35]. Our algorithms will make use of approximation algorithms for various related tree problems.

The *prize-collecting Steiner tree problem* (PCST) is a step stone of our algorithm for the PCSC problem. It was first introduced by Balas [5], and the first constant-approximation algorithm was given by Bienstock et al. [7]. In Goemans and Williamson [19] designed a primal-dual scheme (known as the GW-algorithm) to achieve approximation ratio at most 2. Note that the natural LP formulation of the PCST problem has integrality gap 2. In Archer et al. [1] successfully broke the barrier and presented an algorithm with approximation ratio less than 1.9672.

Our algorithm for the PSC problem is based on an approximation algorithm for the *minimum tree spanning k-vertices problem* (k-MST). The first non-trivial approximation algorithm for k-MST was given by Ravi et al. [33], achieving approximation ratio $O(\sqrt{k})$. After a series of improvements [2, 3, 6, 8, 12, 17], Gary [18] finally obtained a 2-approximation.

The *budgeted spanning tree problem* (BST) is a step stone of our algorithm for the BSC problem. Form the beginning of this century, the approximation ratios for the BST problem were reduced from $(5+\epsilon)$ [23] to $(4+\epsilon)$ [24], and recently to 2 in [32].

1.2 Our contributions

Our contributions are summarized as follows:

- We propose a new sweep cover problem, the *partial sweep cover problem* (PSC). In many applications, it is wasteful to satisfy the coverage requirement of every PoI. Sweep covering only a part of PoIs might be more economic.
- A framework is proposed to solve the mentioned sweep cover problems. Applying the framework, we obtain an 8-approximation algorithm for PSC. For PCSC, we obtain a *lagrangian multiplier preserving algorithm with factor 8* (8-LMP), which is a stronger result than 8-approximation. This result answers the open question proposed in [27], asking whether there exists a constant-approximation algorithm for PCSC without base stations. As for BSC, all previous works either focus on special settings or are heuristic algorithms. We design the first perfor-

mance-guaranteed algorithm in a general case, obtaining a bicriteria $(4, \frac{1}{2})$-algorithm, that is, no more than $4N$ mobile sensors can be used to sweep-cover at least $\frac{1}{2}opt$ PoIs, where N is the budgeted number of mobile sensors and *opt* is the optimal value of BSC. It should be noted that the above algorithms for PSC and BSC can be extended to their weighted versions WPSC and WBSC, maintaining the same bounds of approximation ratios. We believe that our framework may work for more sweep cover problems.

In previous studies on the sweep cover problems, grouping sensors is done by approximating a *corresponding forest problem*. For example, for the MinSSC problem, the number of groups q is guessed, and to group sensors into q groups, a minimum weight forest on the PoIs consisting of exactly q components is constructed. Those PoIs in a same component are assigned to a same group of sensors.

However, such a strategy does not work for the sweep cover problems studied in this paper. For example, for the PSC problem, a natural idea to divide sensors into q groups is to find a *minimum forest on the PoIs with q components spanning at least k PoIs* (k-MSF$_q$). However, there is no known algorithm for k-MSF$_q$, even in terms of approximation. Note that when $q = 1$, k-MSF$_1$ is the classic *minimum tree spanning k vertices* (k-MST) problem which has a 2-approximation. Inspired by the observation that a minimum spanning forest consisting of q components can be obtained from a *minimum spanning tree* by deleting the most costly $q - 1$ edges, a natural idea is to construct a k-MSF$_q$ solution by a similar strategy from an approximate solution of the k-MST problem. The example in Fig. 1 shows that such an idea does not work. In this example, suppose $k = 4$ and $q = 2$. An optimal 4-MST is the path consisting of the rightmost 4 vertices, and deleting the rightmost edge results in a 4-MSF$_2$ solution with weight $n + 1$ (see the middle figure). An optimal 4-MSF$_2$ solution is marked by the thick edges in the third figure, whose weight is 2. So, the approximation ratio is at least $(n + 1)/2$. Besides, the work in [27] shows that it is a challenging task to design an efficient algorithm to find a performance-guaranteed *prize-collecting forest with q components* without specifying roots. Similar challenge also exists for the budgeted forest problem with a specified number of components.

In this paper, we use a different idea to group sensors: construct a *truncation graph*, find a tree corresponding to the problem under consideration in the truncation

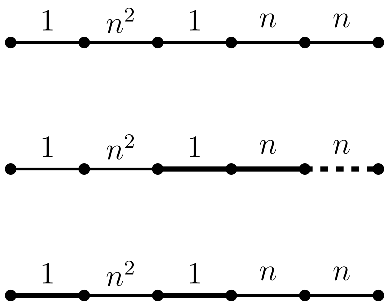

Fig. 1 An example showing that the naive idea of constructing a k-MSF$_q$ solution from a k-MST solution does not work

graph, and then delete those "long" edges to split the tree into a forest. By working on the truncation graph instead of the original graph, we successfully obtained the above declared results.

The rest of this paper is organized as follows. We formally define the three sweep cover problems in Sect. 2 and introduce some preliminary concepts. Section 3 introduces the framework of the unified approach, presents algorithms for PSC (WPSC), PCSC and BSC (WBSC), and gives theoretical analysis of their approximation ratios. Section 4 concludes the paper.

2 Problem formulation and preliminaries

In this section, we give formal definitions of those sweep cover problems studied in this paper and introduce some preliminary results.

Definition 2.1 (sweep cover) Given a graph $G = (V, E)$, a real number t called *sweep-period*, for a set of mobile sensors moving along the edges of G at speed a, a vertex v is said to be *sweep-covered* if in every time period t, it is visited by at least one mobile sensor.

In the following problems, the goals are alway designing trajectories for a set of mobile sensors $\mathcal{S}$ to meet some constraints and achieve some objectives.

Definition 2.2 (partial sweep cover (PSC)) Given a graph $G = (V, E)$ with metric edge length $l : E \mapsto \mathbb{Q}^+$, an integer $k \leq |V|$ and sweep-period t, the goal of PSC is to minimize the number of mobile sensors to sweep-cover at least k vertices.

Definition 2.3 (weighted partial sweep cover (WPSC)) Given G, l, t as the above, together with a vertex weight function $w : V \mapsto \mathbb{Q}^+$ and a number $K \in \mathbb{Q}^+$, the goal of WPSC is to minimize the number of mobile sensors such that the total weight of those vertices that are sweep-covered is at least K, where the weight of a vertex set V' is $w(V') = \sum_{v \in V'} w(v)$.

Definition 2.4 (prize-collecting sweep cover (PCSC)) Given G, l, t as the above, each vertex $v \in V$ is associated with a penalty π_v, PCSC aims to design trajectories for a set $\mathcal{S}$ of mobile sensors to minimize the *cost-plus-penalty value*, that is, $\min\{c \cdot |\mathcal{S}| + \pi(V \backslash \mathcal{C}(\mathcal{S}))\}$, where $\mathcal{C}(\mathcal{S})$ is the set of vertices sweep-covered by the mobile sensors in $\mathcal{S}$, $\pi(V \backslash \mathcal{C}(\mathcal{S})) = \sum_{v \notin \mathcal{C}(\mathcal{S})} \pi_v$ is the penalty on the uncovered vertices and c is the cost of a mobile sensor.

Definition 2.5 (budgeted sweep cover (BSC)) Given G, l, t as the above and a positive integer N, the goal of BSC is to use N mobile sensors to sweep-cover the maximum number of vertices.

Definition 2.6 (weighted budgeted sweep cover (WBSC)) Given G, l, t as the above, together with a vertex weight function $w : V \mapsto \mathbb{Q}^+$ and a positive integer N, the goal of WBSC is to use at most N mobile sensors to sweep-cover a subset of vertices such that the total weight of those sweep-covered vertices is maximized.

For a positive real number $r \geq 1$ (resp. $r \leq 1$), a polynomial time algorithm for a minimization (resp. maximization) problem is said to be an *r-approximation algorithm* if for any instance I of the problem, the output of the algorithm has value at most (resp. at least) r times that of an optimal solution. If r can be achieved by some instance, then it is called the *approximation ratio* of the algorithm.

An *r-Lagrangian Multiplier Perserving* (r-LMP) algorithm is stronger than an r-approximation algorithm. For a real number $r \geq 1$, an algorithm for PCSC is said to be an r-LMP if for any PCSC instance I, it can always compute in polynomial time a solution $\mathcal{S}$ (and determine the routes for those mobiles sensors in $\mathcal{S}$) satisfying

$$c \cdot |\mathcal{S}| + r \cdot \pi(V \setminus \mathcal{C}(\mathcal{S})) \leq r \cdot opt(I),$$

where $opt(I)$ is the optimal value of instance I.

A bicriteria approximation algorithm allows the constraint to be violated, but "no too much", while the approximation ratio can be guaranteed. For real numbers $\beta \geq 1$ and $0 < \alpha \leq 1$, an algorithm for BSC is said to be a (β, α)-bicriteria approximation if for any BSC instance I, it can always compute in polynomial time a solution $\mathcal{S}$ (and determine the routes for those mobiles sensors in $\mathcal{S}$) satisfying

$$|\mathcal{S}| \leq \beta \cdot N \text{ and } |\mathcal{C}(\mathcal{S})| \geq \alpha \cdot opt(I),$$

where $opt(I)$ is the optimal value of instance I.

To solve the above problems, our algorithms make use of approximation algorithms for *corresponding tree problems*, the goals of which are to find trees with similar constraints and similar objectives as the sweep cover problems.

The PSC problem corresponds to the k-MST problem.

Definition 2.7 (minimum tree spanning k vertices (k-MST)) Given a graph $G = (V, E)$ with edge length $l : E \mapsto \mathbb{Q}^+$ and an integer $k \leq |V|$, the goal of k-MST is to find a minimum length tree that spans at least k vertices, where the length of a tree T is $l(T) = \sum_{e \in E(T)} l(e)$.

The WPSC problem corresponds to the QMST problem.

Definition 2.8 (quota minimum spanning tree (QMST)) Given G, l as the above, vertex weight $w : V \mapsto \mathbb{Q}^+$ and a quota $K \in \mathbb{Q}^+$, the goal of the QMST problem is to find a minimum length tree T satisfying $w(T) = \sum_{v \in V(T)} w(v) \geq K$.

The PCSC problem corresponds to the PCST problem.

Definition 2.9 (prize-collecting Steiner tree (PCST)) Given G, l as the above, each vertex $v \in V$ is associated with a penalty π_v, the goal of PCST is to find a tree T such that the *length-plus-penalty value* is minimized, that is, $\min\{l(T) + \pi(V \setminus V(T)\}$, where $\pi(V \setminus V(T)) = \sum_{v \notin V(T)} \pi_v$.

The BSC problem corresponds to the BST problem.

Definition 2.10 (budgeted spanning tree (BST)) Given G, l as the above and a budget $B \in \mathbb{Q}^+$, the goal of BST is to find a tree T containing the maximum number vertices subjected to the constraint $l(T) \leq B$.

The WBSC problem corresponds to the WBST problem.

Definition 2.11 (weighted budgeted spanning tree (WBST)) Given G, l, t as the above, a vertex weight function $w : V \mapsto \mathbb{Q}^+$ and a budget $B \in \mathbb{Q}^+$, the goal of WBST is to find a tree T with $l(T) \leq B$ such that $w(T) = \sum_{v \in V(T)} w(v)$ is maximized.

3 Algorithms and their theoretical analysis

In this section, we first introduce the framework of the unified approach. Then, we apply it to PSC (WPSC), PCSC and BSC (WBSC), respectively, and theoretically analyze their approximation ratios.

3.1 Framework of the unified approach

The framework for a sweep cover problem consists of two steps: *vertex grouping step* and *sensor allocation step*. A *forest* is obtained in the first step. The vertices in one component of the forest are said to be in a common *group* and will be sweep-covered by a same set of mobile sensors. In the sensor allocation step, each component T_i of the forest is transformed into a cycle C_i using a method similar to that of constructing a 2-approximate solution to the TSP problem [34]. By the metric assumption on l, we have

$$l(C_i) \leq 2l(T_i) \text{ for each } i. \tag{1}$$

Then mobile sensors are deployed uniformly along the cycles, as shown in line 6 to line 10 of Algorithm 1. Such a scheduling forms a sweep cover for $V(C_i)$'s.

The crucial step is how to construct a forest to group vertices. Our method is to construct a *truncation graph* G^t with the same topology as G and a truncated edge length function l^t defined as follows:

$$l^t(e) = \begin{cases} l(e)/at, & \text{if } l(e) \leq at, \\ 1, & \text{if } l(e) > at. \end{cases} \tag{2}$$

Find a tree T^t in G^t (using different algorithm $\mathcal{A}_{tree}$ according to different sweep cover problem under consideration), which corresponds to a tree T in the original graph G. Delete those edges from T whose lengths are larger than at, resulting in a forest. The details of the unified framework is presented in Algorithm 1.

Algorithm 1 A unified algorithm for a sweep cover instance I_{SC}

Input: A metric graph G.

Output: A schedule of routes for mobile sensors.

```
T^t ← a tree of the truncation graph G^t by calling algorithm A_tree
T ← the tree of G corresponding to T^t.
T_1, ... T_q ← subtrees obtained from T by deleting edges with lengths larger than at.
for i = 1, ... q do
    construct cycle C_i from T_i satisfying l(C_i) ≤ 2l(T_i).
    if |V(C_i)| = 1 then
        deploy a mobile sensor to provide continuous coverage for V(C_i).
    else
        uniformly deploy ⌈l(C_i)/at⌉ mobile sensors on C_i and let them move along the
        cycle in a same direction at speed a.
    end if
end for
return the above schedule.
```

3.2 Partial sweep cover

For the PSC problem, algorithm $\mathcal{A}_{tree}$ is taken to be an approximation algorithm for the k-MST problem on the truncated instance (G^t, l^t).

Theorem 3.1 *If Algorithm* 1 *employs an α-approximation algorithm for the truncated k-MST instance, then it computes a* 4α*-approximation for PSC.*

Proof Let opt_{PSC} be the optimal value of a PSC instance. We first trim the trajectories of the opt_{PSC} mobile sensors into a feasible solution to the k-MST instance on the truncation graph G^t. The trajectory of each mobile sensor in time span $[0, t]$ is a walk in G with length at. Let H be the union of these walks, and let H^t be the subgraph of G^t corresponding to H. Note that H contains at least k vertices, and so does H^t,. Furthermore, $l(H) \leq opt_{PSC} \cdot at$. Then by the definition of l^t in (2), we have

$$l^t(H^t) \leq l(H)/at \leq opt_{PSC}. \tag{3}$$

Note that H^t contains at most opt_{PSC} connected components. Hence adding at most $opt_{PSC} - 1$ edges to H^t can transform it into a connected subgraph H_c^t. Let T_c^t be a spanning tree of H_c^t. Then, T_c^t is a feasible solution to the k-MST instance. Furthermore, by (3) and the observation that edges in G^t have lengths at most 1, we have

$$l^t(T_c^t) \leq l^t(H_c^t) \leq l^t(H^t) + opt_{PSC} - 1 \leq 2opt_{PSC} - 1. \tag{4}$$

Let $\mathcal{S}_p$ be the set of mobile sensors computed by Algorithm 1. Next, we estimate $|\mathcal{S}_p|$. Note that each group C_i is deployed at most $l(C_i)/at + 1$ mobile sensors.

First consider the case that $q = 1$. In this case, all the edges of T in line 2 of the algorithm have lengths at most at, and thus by the definition of l^t,

$$|\mathcal{S}_p| \le \frac{l(C_1)}{at} + 1 \le \frac{2l(T_1)}{at} + 1 = 2l^t(T^t) + 1 \le 2\alpha \cdot l^t(T_c^t) + 1 < 4\alpha \cdot opt_{PSC}, \quad (5)$$

where the second inequality comes from (1), the third inequality holds because T^t is an α-approximate solution to the k-MST instance on G^t and T_c^t is a feasible solution to the k-MST instance on G^t, and the last inequality comes from (4) and the fact $\alpha \ge 1$.

Next consider the case that $q \ge 2$. Note that $\{T_i\}_{i=1,\dots,q}$ are obtained from T by deleting those edges with lengths larger than at, there are exactly $q-1$ edges that are deleted, and the deleted edges have lengths 1 in G^t. Hence

$$l^t(T^t) = \sum_{i=1}^{q} \frac{l(T_i)}{at} + (q-1).$$

Combining this with a similar argument for (5), and using the assumption $q \ge 2$, we have

$$\begin{aligned} |\mathcal{S}_p| &\le \sum_{i=1}^{q} \left(\frac{l(C_i)}{at} + 1 \right) = \sum_{i=1}^{q} \frac{l(C_i)}{at} + q \le \sum_{i=1}^{q} \frac{2l(T_i)}{at} + 2(q-1) \\ &= 2l^t(T^t) \le 2\alpha \cdot l^t(T_c^t) < 4\alpha \cdot opt_{PSC}. \end{aligned} \quad (6)$$

The desired approximation ratio 4α is proved. □

If $\mathcal{A}_{tree}$ is taken to be Garg's 2-approximation algorithm for the k-MST problem [18], then we have the following approximation ratio.

Corollary 3.2 *PSC admits an* 8*-approximation.*

For the weighted version of the PSC problem, namely WPSC, the algorithm $\mathcal{A}_{tree}$ is taken to be an approximation algorithm for the QMST problem. In Awerbuch et al. [4] proved that a polynomial time approximation algorithm for the k-MST problem can be used to get a pseudopolynomial time approximation algorithm for the QMST problem with the same approximation guarantee. Johnson et al. [23] found that the pseudopolynomial time algorithm can be converted into a truly polynomial time one. Therefore, for QMST, there exists a polynomial time 2-approximation algorithm. Using a similar argument as that for Theorem 3.1, by noting that the trajectories of the set of mobile sensors in an optimal solution can be trimmed into a tree T_c^t in G^t with

$$w(T_c^t) \ge K,$$

we have the following theorem.

Theorem 3.3 *There exists an* 8*-approximation algorithm for WPSC.*

3.3 Prize-collecting sweep cover

For a PCSC instance (G, l, π), let (G^t, l^t, π^t) be a PCST instance on the truncation graph G^t with truncated edge length function l^t and truncated vertex penalty function π^t:

$$\pi^t(v) = \pi(v)/at \ \text{ for any } \ v \in V. \tag{7}$$

The algorithm $\mathcal{A}_{tree}$ is taken to be an r-LMP algorithm for the prize-collecting Steiner tree (PCST) problem. A *scaling technique* similar to that in [27] is employed here: assume

$$\frac{at}{c} = \frac{1}{2}. \tag{8}$$

This can be assumed without loss of generality, because we can scale π and c simultaneously without changing the solution and also keeps the approximation ratio.

Theorem 3.4 *If Algorithm* 1 *employs an r-LMP algorithm for the truncated PCST instance, then it computes a* 4*r-LMP solution to the PCSC problem.*

Proof Suppose an optimal solution to a PCSC instance uses opt_{PC} mobile sensors and denote by Z^* the set of vertices that are not sweep-covered. Then the optimal value is $c \cdot opt_{PC} + \pi(Z^*)$. Similar to the argument for the PSC problem, the trajectories of these opt_{PC} mobile sensors in time span $[0, t]$ can be trimmed into a tree T_c^t in G^t, which spans vertex set $V \backslash Z^*$ and has

$$l^t(T_c^t) \leq 2opt_{PC} - 1. \tag{9}$$

Let $\mathcal{S}_{PC}$ be the set of mobile sensors computed by Algorithm 1, and let Z be the set of un-covered vertices. If $q \geq 2$, then similar to the derivation of (6), we have

$$|\mathcal{S}_{PC}| \leq 2l^t(T^t). \tag{10}$$

It follows that

$$\begin{aligned} c \cdot |\mathcal{S}_{PC}| + 4r \cdot \pi(Z) &\leq c \cdot 2l^t(T^t) + 4rat \cdot \pi^t(Z) \\ &= 2c \cdot (l^t(T^t) + r \cdot \pi^t(Z)) \leq 2rc \cdot (l^t(T_c^t) + \pi^t(Z^*)) \\ &< 2rc \cdot (2opt_{PC} + \pi(Z^*)/at) = 4r(c \cdot opt_{PC} + \pi(Z^*)), \end{aligned}$$

where the first inequality is by (10) and (7), the first equality uses (8), the second inequality holds because T^t is an r-LMP solution to the PCST problem, the last inequality comes from (9) and the definition of π^t, and the last equality also makes use of (8).

If $q = 1$, then

$$|\mathcal{S}_{PC}| \leq 2l^t(T_1^t) + 1.$$

Using a similar argument, we have

$$\begin{aligned}
& c \cdot |\mathcal{S}_{PC}| + 4r \cdot \pi(Z) \\
\leq\ & c \cdot (2l^t(T_1^t) + 1) + 4rat \cdot \pi^t(Z) = 2c \cdot \left(l^t(T_1^t) + \frac{1}{2} + r \cdot \pi^t(Z) \right) \\
\leq\ & 2rc \cdot (l^t(T_c^t) + \pi^t(Z^*) + \frac{1}{2r}) \leq 2rc \cdot \left(2opt_{PC} - 1 + \frac{1}{2r} + \pi(Z^*)/at \right) \\
<\ & 2rc \cdot (2opt_{PC} + \pi(Z^*)/at) = 4r\big(c \cdot opt_{PC} + \pi(Z^*)\big),
\end{aligned}$$

where the last inequality holds because $r \geq 1$. The theorem is proved. □

If $\mathcal{A}_{tree}$ is taken to be the Goemans and Williamson's 2-LMP algorithm for the PCST problem, then we have the following performance.

Corollary 3.5 *PCSC admits an* 8-*LMP*.

3.4 Budgeted sweep cover

In this section, we obtain a $(4, \frac{1}{2})$-bicriteria approximation algorithm for the BSC problem. That is, there exists a sweep cover scheduling that can sweep-cover at least $\frac{1}{2}opt_B$ vertices using at most $4N$ mobile sensors, where opt_B is the maximum number of vertices that can be sweep-covered by N mobile sensors. Given a BSC instance (G, N), let $(G^t, 2N-1)$ be an instance of the budgeted spanning tree (BST) problem on truncation graph G^t asking for a tree of length at most $2N-1$ to span the maximum number of vertices.

Theorem 3.6 *If Algorithm* 1 *employs an* α*-approximation algorithm for the truncated BST instance*, *then it computes a* $(4, \alpha)$*-bicriteria solution to the BSC problem.*

Proof Let $\mathcal{S}_B$ be the set of mobile sensors computed by Algorithm 1. Similar to the derivation of (5) and (6), we have $|\mathcal{S}_B| \leq 2l^t(T^t)$ for $q \geq 2$ and $|\mathcal{S}_B| \leq 2l^t(T^t) + 1$ for $q = 1$. Because the budget for the truncated BST instance is $2N-1$, we have

$$|\mathcal{S}_B| \leq 2l^t(T^t) + 1 < 4N.$$

That is, the budge for the BSC instance is violated by at most 4 times.

Next we estimate the number of sweep-covered vertices. Let opt_B be the number of vertices sweep-covered by an optimal scheduling. Similar to the construction of T_c^t in the proof of Theorem 3.1, the trajectories of those mobile sensors in an optimal solution can be trimmed into a tree T_c^t of G^t with

$$|V(T_c^t)| \geq opt_B \ \text{ and } \ l(T_c^t) \leq 2N - 1.$$

Hence T_c^t is a feasible solution to the truncated BST instance spanning opt_B vertices. Because T^t is an α-approximate solution to the truncated BST instance, we have

$$|V(T^t)| \geq \alpha \cdot |V(T_c^t)| \geq \alpha \cdot opt_B.$$

The approximation ratio α follows. □

If $\mathcal{A}_{tree}$ is taken to be the Paul's $\frac{1}{2}$-approximation algorithm for the BST problem [32], then we have the following result.

Corollary 3.7 *BSC admits a* $(4, \frac{1}{2})$*-bicriteria approximation.*

Paul et al. had extended their $\frac{1}{2}$-approximation algorithm for the BST problem to its weighted version WBST, achieving the same lower bound of the approximation ratio. Making using of this algorithm, the weighted BSC problem, namely the WBSC problem, also admits a bicriteria approximation algorithm with the same performance.

Theorem 3.8 *There exists a* $(4, \frac{1}{2})$*-bicriteria approximation algorithm for WBSC.*

4 Conclusion and future work

Three types of sweep cover problems are considered in this paper: the partial sweep cover problem PSC, the prize-collecting sweep cover problem PCSC, and the budgeted sweep cover problem BSC. A unified algorithm works for all of them, achieving 8-approximation for PSC, 8-LMP for PCSC, and bicriteria $(4, \frac{1}{2})$-approximation for BSC. The PSC problem is initiated in this paper. The second result answers a question proposed in [27]. And the third result gives the first theoretically guaranteed performance for the BSC problem in a general setting.

In a sweep cover problem, how to group sensors is a crucial step. In many previous works, this was done by fining a forest with q components, where q is the guessed number of groups. As implied by the work in [27], this is a challenging task if what we need is not a "spanning" forest. In this paper, we circumvent this difficulty by first finding a tree (instead of a forest) in a truncation graph, and then transforming it into a forest by deleting "long" edges. We believe this framework might be useful for some other sweep cover problems.

As for the BSC problem, what is obtained in this paper is only a bicriteria result. How to obtain an approximation algorithm without violation is worth further studying.

Acknowledgements This research is supported by National Natural Science Foundation of China (U20A2068), Zhejiang Provincial Natural Science Foundation of China (LD19A010001) and NSF of USA under Grant III-1907472.

References

1. Archer, A., Bateni, M.H., Hajiaghayi, M.T., Karloff, H.: Improved approximation algorithms for prize-collecting steiner tree and tsp. SIAM J. Comput. **40**(2), 309–332 (2011)
2. Arora, S., Karakostas, G.: A 2+ϵ approximation algorithm for the k-mst problem. Math. Program. **107**(3), 491–504 (2006)
3. Arya, S., Ramesh, H.: A 2.5-factor approximation algorithm for the k-mst problem. Inf. Process. Lett. **65**(3), 117–118 (1998)
4. Azar, B.A.Y., Blum, A., Vempala, S.: New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. SIAM J. Comput. **28**(1), 254–262 (1998)
5. Balas, E.: The prize collecting traveling salesman problem. Networks **19**(6), 621–636 (1989)
6. Baruch, A., Yossi, A., Avrim, B., Vempala, S.: Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. SIAM J. Comput. **28**(1), 254–262 (1999)
7. Bienstock, D., Goemans, M.X., Simchi-Levi, D., Williamson, D.: A note on the prize collecting traveling salesman problem. Math. Program. **59**(1), 413–420 (1993)
8. Blum, A., Ravi, R., Vempala, S.: A constant-factor approximation algorithm for the k mst problem. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 442–448 (1996)
9. Chen, Z., Zhu, X., Gao, X., Wu, F., Gu, J., Chen, G.: Efficient scheduling strategies for mobile sensors in sweep coverage problem. In: 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–4 (2016)
10. Chen, Qingqing, Huang, Xiaohui, Ran, Yingli: Approximation algorithm for distance constraint sweep coverage without predetermined base stations. Discrete Math. Algorithms Appl. **10**(05), 1850064 (2018)
11. Cheng, W., Li, M., Liu, K., Liu, Y., Li, X., Liao, X.: Sweep coverage with mobile sensors. In: 2008 IEEE International Symposium on Parallel and Distributed Processing, pp. 1–9 (2008)
12. Chudak, F.A., Roughgarden, T., Williamson, D.P.: Approximate k-msts and k-steiner trees via the primal-dual method and lagrangean relaxation. Math. Program. **100**(2), 411–421 (2004)
13. Feng, Y., Gao, X., Wu, F., Chen, G.: Shorten the trajectory of mobile sensors in sweep coverage problem. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2015)
14. Gage, D.W.: Command Control for Many-robot Systems. Technical Report, Naval Command Control and Ocean Surveillance Center Rdt and E Div San Diego CA (1992)
15. Gao, X., Fan, J., Wu, F., Chen, G.: Cooperative sweep coverage problem with mobile sensors. IEEE Trans. Mob. Comput 1–1 (2020)
16. Gao, X., Fan, J., Fan, W., Chen, G.: Approximation algorithms for sweep coverage problem with multiple mobile sensors. IEEE/ACM Trans. Netw. **26**(2), 990–1003 (2018)
17. Garg, N.: A 3-approximation for the minimum tree spanning k vertices. In: Proceedings of 37th Conference on Foundations of Computer Science. pp. 302–309. IEEE (1996)
18. Garg, N.: Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, pp. 396–402 (2005)
19. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. SIAM J. Comput. **24**(2), 296 (1995)
20. Gorain, B., Mandal, P.S.: Approximation algorithm for sweep coverage on graph. Inf. Process. Lett. **115**(9), 712–718 (2015)
21. Gorain, B., Mandal, P.S.: Solving energy issues for sweep coverage in wireless sensor networks. Discrets Appl. Math. **228**, 130–139 (2017)
22. Huang, P., Zhu, W., Liao, K., Sellis, T., Zhiyong, Y., Guo, L.: Efficient algorithms for flexible sweep coverage in crowdsensing. IEEE Access **6**, 50055–50065 (2018)
23. Johnson, D.S., Minkoff, M., & Phillips, S.: The prize collecting steiner tree problem: theory and practice. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '00, pp. 760–769. Society for Industrial and Applied Mathematics (2000)
24. Levin, A.: A better approximation algorithm for the budget prize collecting tree problem. Oper. Res. Lett. **32**(4), 316–319 (2004)
25. Li, M., Cheng, W., Liu, K., He, Y., Li, X., Liao, X.: Sweep coverage with mobile sensors. IEEE Trans. Mob. Comput. **10**(11), 1534–1545 (2011)
26. Liang, D., Shen, H.: Chargeable sweep coverage problem. arXiv e-prints, arXiv:2105.06030 (2021)

27. Liang, W., Zhang, Z.: Approximation algorithm for prize-collecting sweep cover with base stations. Theor. Comput. Sci., a preliminary version was presented in AAIM2020 (2022)
28. Liang, D., Shen, H.: Efficient algorithms for max-weighted point sweep coverage on lines. Sensors (Basel, Switzerland) **21**(4), 1457 (2021)
29. Liang, J., Huang, X., Zhang, Z.: Approximation algorithms for distance constraint sweep coverage with base stations. J. Comb. Optim. **37**(4), 1111–1125 (2019)
30. Liu, C., Hongwei, D., Ye, Q., Wen, X.: Group sweep coverage with guaranteed approximation ratio. Theoret. Comput. Sci. **836**, 1–15 (2020)
31. Nie, Z., Hongwei, D.: An approximation algorithm for general energy restricted sweep coverage problem. Theoret. Comput. Sci. **864**, 70–79 (2021)
32. Paul, A., Freund, D., Ferber, A., Shmoys, D.B., Williamson, D.P.: Budgeted prize-collecting traveling salesman and minimum spanning tree problems. Math. Oper. Res. **45**(2), 576–590 (2020)
33. Ravi, R., Sundaram, R., Marathe, M.V., Rosenkrantz, D.J., Ravi, S.S.: Spanning trees–short or small. SIAM J. Discrete Math. **9**(2), 178–200 (1996)
34. Vazirani, V.V.: Approximation Algoritms. Springer, New York (2001)
35. Wu, W., Zhang, Z., Lee, W., Du, D.: Optimal Coverage in Wireless Sensor Networks. Springer, New York (2020)
36. Yaoyao, G., Bozda, D., Brewer, R.W., Ekici, E.: Data harvesting with mobile elements in wireless sensor networks. Comput. Netw. **50**(17), 3449–3465 (2006)
37. Zhao, D., Ma, H., Liu, L.: Mobile sensor scheduling for timely sweep coverage. In: IEEE Wireless Communications and Networking Conference, WCNC, pp. 1771–1776, 04 (2012)
38. Zixiong, N., Chuang, L., Hongwei, D.: Data sensing with limited mobile sensors in sweep coverage. In: International Conference on Combinatorial Optimization and Applications, pp. 669–680. Springer (2020)