Approximation Algorithms for the Connected Sensor Cover Problem

Lingxiao Huang Jian Li * Qicai Shi
Institute for Interdisciplinary Information Sciences
Tsinghua University, China
{huanglx12@mails,lijian83@mail,sqc12@mails}.tsinghua.edu.cn

January 24, 2017

Abstract

We study the minimum connected sensor cover problem (MIN-CSC) and the budgeted connected sensor cover (Budgeted-CSC) problem, both motivated by important applications (e.g., reduce the communication cost among sensors) in wireless sensor networks. In both problems, we are given a set of sensors and a set of target points in the Euclidean plane. In MIN-CSC, our goal is to find a set of sensors of minimum cardinality, such that all target points are covered, and all sensors can communicate with each other (i.e., the communication graph is connected). We obtain a constant factor approximation algorithm, assuming that the ratio between the sensor radius and communication radius is bounded. In Budgeted-CSC problem, our goal is to choose a set of B sensors, such that the number of targets covered by the chosen sensors is maximized and the communication graph is connected. We also obtain a constant approximation under the same assumption.

1 Introduction

In many applications, we would like to monitor a region or a collection of targets of interests by deploying a set of wireless sensor nodes. A key challenge in such applications is the limited energy supply for each sensor node. Hence, designing efficient algorithms for minimizing energy consumption and maximizing the lifetime of the network is an important problem in wireless sensor networks and many variations have been studied extensively. We refer interested readers to the book by Du and Wan [11] for many algorithmic problems in this domain.

In this paper, we consider two important sensor coverage problems. Now, we introduce some notations and formally define our problem. We are given a set $\mathcal{S}$ of n sensors in $\mathbb{R}^d$. All sensors in $\mathcal{S}$ have the same communication range R_{c} and the same sensing range R_{s}. In other words, two sensors s and s' can communicate with each other if $\mathsf{dist}(s, s') \leq R_{\mathsf{c}}$, and a target point p can be covered by sensor s if $\mathsf{dist}(p, s) \leq R_{\mathsf{s}}$. We use $D(s, R)$ to denote the disk with radius R centered at point s. Let $D_{\mathsf{c}}(s) = D(s, R_{\mathsf{c}})$ and $D_{\mathsf{s}}(s) = D(s, R_{\mathsf{s}})$.

Assumption 1 *In this paper, we assume that $C = \max\{1, R_{\mathsf{s}}/R_{\mathsf{c}}\}$ is a constant. Note that this assumption holds for most practical applications. Without loss of generality, we can assume that $R_{\mathsf{c}} = 1$. Hence, $R_{\mathsf{s}} = O(1)$.*

*Corresponding author.

The first problem we study is the the *minimum Connected sensor covering* (MIN-CSC) problem. This problem considers the problem of selecting the minimum number of sensors that form a connected network and detect all the targets. It is somewhat similar, but different from, the connected dominating set problem. We will discuss the difference shortly. The formal problem definition is as follows:

Definition 1 MIN-CSC*: Given a set $\mathcal{S}$ of sensors and a set $\mathcal{P}$ of target points, find a subset $\mathcal{S}' \subseteq \mathcal{S}$ of minimum cardinality such that all points in $\mathcal{P}$ are covered by the union of sensor areas in $\mathcal{S}'$ and the communication links between sensors in $\mathcal{S}'$ form a connected graph.*

In some applications, instead of monitoring a set of discrete target points, we would like to monitor a continuous range R, such as a rectangular area. Such problems can be easily converted into a MIN-CSC with discrete points, by creating a target point (which we need to cover) in each cell of the arrangement of the sensing disks $\{D_{\mathsf{s}}(s)\}_{s\in\mathcal{S}}$ restricted in R.

The second problem studied in this paper is the *Budgeted connected sensor cover* (Budgeted-CSC) problem. The problem setting is the same as MIN-CSC, except that we have an upper bound on the number of sensors we can open, and the goal becomes to maximize the number of covered targets.

Definition 2 Budgeted-CSC*: Given a set $\mathcal{S}$ of sensors , a set $\mathcal{P}$ of target points and a positive integer B, find a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'| \leq B$ and the number of points in $\mathcal{P}$ covered by the union of sensor areas in $\mathcal{S}'$ is maximum and the communication links between sensors in $\mathcal{S}'$ form a connected graph.*

1.1 Previous Results and Our Contributions

1.1.1 MIN-CSC

The MIN-CSC problem was first proposed by Gupta et al. [19]. They gave an $O(r \ln n)$-approximation (r is an upper bound of the hop-distance between any two sensors having nonempty sensing intersections). Wu et al. [37] give an $O(r)$-approximation algorithm, which is best approximation ratio known so far (in terms of r). If $R_{\mathsf{s}} \leq R_{\mathsf{c}}/2$, $r = 1$ and the above result implies a constant approximation. However, even if R_{s} is slightly larger than $R_{\mathsf{c}}/2$, r may still be arbitrarily large.

MIN-CSC is in fact a special case the *group Steiner tree* problem (as also observed in Wu et al [37]). In fact, this can be seen as follows: consider the communication graph (the edges are the communication links). For each target, we create a group which consists for all sensor nodes that can cover the target. The goal is to find a minimum cost tree spanning all groups. [1] Garg et al [16], combined with the optimal probabilistic tree embedding [13], obtained an $O(\log^3 n)$ factor approximation algorithm the group Steiner tree problem via LP rounding. Chekuri et al. [6] obtained nearly the same approximation ratio using pure combinatorial method.

Our first main contribution is a constant factor approximation algorithm for MIN-CSC under Assumption 1, improving on the aforementioned results. Our improvement heavily rely on the geometry of the problem (which the group Steiner tree approach ignores).

Theorem 1 *There is a polynomial time approximation algorithm which can achieve an approximation factor $O(C^2)$ for* MIN-CSC*. Under Assumption 1, the approximation factor is a constant.*

[1]Notice that the group Steiner tree is edge-weighted but MIN-CSC is node-weighted. However, since all nodes have the same (unit) weight, the edge-weight and node-weight of a tree differ by at most 1.

1.1.2 Budgeted-CSC

Recall in Budgeted-CSC, we have a budget B, which is the upper bound of the number of sensors we can use and our goal is to maximize the number of covered target points. Kuo et al.[26] study this problem under the assumption that the communication and the sensing radius of sensors are the same (i.e., $R_s = R_c$). They obtained an $O(\sqrt{B})$-approximation by transforming the problem to a more general connected submodular function maximization problem.

Recently, Khuller et al. [24] obtained a constant approximation for the *budgeted generalized connected dominating set problem*, defined as follows: Given an undirected graph $G(V, E)$ and budget B, and a monotone *special submodular function* [2] $f : 2^V \to \mathbb{Z}^+$, find a subset $S \subseteq V$ such that $|S| \leq B$, S induces a connected subgraph and $f(S)$ is maximized. If $R_s \leq R_c/2$ in Budgeted-CSC, the coverage function $f(S)$ (the number of targets covered by sensor set S) is a special submodular function. Hence, we have a constant approximation for Budgeted-CSC when $R_s \leq R_c/2$. When $R_s > R_c/2$, $f(S)$ may not be special submodular and the algorithm and analysis in [24] do not provide any approximation guarantee for Budgeted-CSC.

We note that it is also possible to adapt the greedy approach developed by group Steiner tree [6] and polymatroid Steiner tree [4] to get polylogarithmic approximation for Budgeted-CSC. However, it is unlikely that the approach can be made to achieve constant approximation factors, and we omit the details.

In this paper, we improve the above results by presenting the first constant factor approximation algorithm under the more general Assumption 1.

Theorem 2 *There is a polynomial time approximation algorithm which can achieve approximation factor of* $\frac{1}{102C^2}$ *for* Budgeted-CSC. *Under Assumption 1, the approximation factor is* $O(1)$.

Our algorithm is inspired by [24]. In particular, we make crucial use of the geometry of the problem to get around the issue required by [24] (i.e., the coverage function is required to be special submodular in their work).

1.2 Other Related Work

MIN-CSC is closely related to the *minimum dominating set* (MIN-DS) problem and the *minimum connected dominating set* (MIN-CDS) problem. In fact, if the communication radius R_c is the same as the sensing radius R_s, MIN-CSC reduces to (unweighted) MIN-CDS. In general graphs, MIN-CDS inherits the inapproximability of set cover, so it is NP-hard to approximation MIN-CDS within a factor of $\rho \ln n$ for any $\rho < 1$ [14, 10]. Improving upon Klein et al. [25], Guha et al.[18] obtained a $1.35 \ln n$-approximation, which is the best result known for general graphs.

Lichtenstein et al. [28] proved that MIN-CDS in unit disk graphs (UDG) is NP-hard (which also implies that MIN-CSC is NP-hard). The first constant approximation algorithm for the unweighted MIN-CDS problem in UDG was obtained by Wan et al.[34]. This was later improved by Cheng et al.[7], who gave the first PTAS. For the weighted (connected) dominating set problem , Ambühl et al. [1] obtained the first constant ratio approximation algorithms for both problems (the constants are 72 and 94 for MIN-DS and MIN-CDSrespectively). The constants were improved in a series of subsequent papers [21, 9, 39, 36]. Very recently, Li and Jin [27] obtained the first PTAS for weighted MIN-DS and an improved constant approximation for weighted MIN-CDS in UDG. Many variants of MIN-DS and MIN-CDS, motivated by various applications in wireless sensor network, have been studied extensively. See [11] for a comprehensive treatment.

[2] f is a special submodular function if (1) f is submodular: $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ for any $A \subset B \subseteq V$; (2) $f(A \cup X) - f(A) = f(A \cup B \cup X) - f(A \cup B)$ if $N(X) \cap N(B) = \emptyset$ for any $X, A, B \subseteq V$. Here, $N(X)$ denotes the neighborhood of X (including X).

Budgeted-CSC is a special case of the submodular function maximization problem subject to a cardinality constraint and a connectivity constraint. Submodular maximization under cardinality constraint, which generalizes the maximum coverage problem, is a classical combinatorial optimization problem and it is known the optimal approximation is $1 - 1/e$ [30, 14]. Submodular maximization under various more general combinatorial constraints (in particular, downward monotone set systems) is a vibrant research area in theoretical computer science and there have been a number of exciting new developments in the past few years (see e.g., [3, 33] and the references therein). The connectivity constraint has also been considered in some previous work [38, 26, 24], some of which we mentioned before.

2 Preliminaries

We need the following *maximum coverage* (MaxCov) in our algorithms.

Definition 3 MaxCov*: Given a universe U of elements and a family $\mathcal{S}$ of subsets of U, and a positive integer B, find a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'| \leq B$ and the number of elements covered by $\cup_{S \in \mathcal{S}'} S$ is maximized.*

We need to following well known result, by [30, 20].

Lemma 1 (Corollary 1.1 of Hochbaum et al. [20]) *The greedy algorithm is a $(1 - \frac{1}{e})$-approximation for* MaxCov.

A closely related problem is the *hitting set* problem.

Definition 4 HitSet*: Given a universe U of weighted elements (with weight function $c : U \to \mathbb{R}^+$) and a family $\mathcal{S}$ of subsets of U find a subset $H \subseteq U$ such that $H \cap S \neq \emptyset$ for all $S \in \mathcal{S}$ (i.e., H hits every subset in $\mathcal{S}$) and $\sum_{u \in H} c_u$ is minimized.*

The HitSet problem is equivalent to the set cover problem (where the elements and subsets switch roles). It is well known that a simple greedy algorithm can achieve an approximation factor of $\ln n$ for HitSet and the factor is essentially optimal [14, 10]. In this paper, we use a geometric version of HitSet in which the set of given elements are points in $\mathbb{R}^2$ and the subsets are induced by given disks (i.e., each $S \in \mathcal{S}$ is the subset of points that can be covered by a given disk). Geometric hitting set admits constant factor approximation algorithms (even PTAS) for many geometric objects (including disks) [2, 8, 29, 32, 5]. As mentioned in the introduction, MIN-CSC is a special case of the following *group Steiner tree* (GST) problem.

Definition 5 GST*: We are given an undirected graph $G = (V, E, c, \mathcal{F})$ where $c : E \to \mathbb{Z}^+$ is the edge cost function, and $\mathcal{F}$ is a collection of subsets of V. Each subset in $\mathcal{F}$ is called a group. The goal is to find a subtree T, such that $T \cap S \neq \emptyset$ for all $S \in \mathcal{F}$ (i.e., T spans all groups) and the cost of the tree $\sum_{e \in T} c_e$ is minimized.*

Our algorithm for Budgeted-CSC also needs the following *quota Steiner tree* (QST) problem.

Definition 6 QST*: Given an undirected graph $G = (V, E, c, p)$ ($c : E \to \mathbb{Z}^+$ is the edge cost function, $p : V \to \mathbb{Z}^+$ is the vertex profit function) and an integer q, find a subtree $T = \arg\min_{T \subset E, \sum_{v \in T} p(v) \geq q} \sum_{e \in T} c(e)$ of the graph G (T tries to collect as much profit as possible subject to the quota constraint).*

Johnson et al. [22] proposed the QST problem and proved that any α-approximation for the k-MST problem yields an α-approximation for the QST problem. Combining with the 2-approximation for $k-$MST developed by Garg [15], we can get a 2-approximation for the QST problem.

Lemma 2 *These is an approximation algorithm with approximation factor 2 for* QST.

3 Minimum Connected Sensor Cover

We first construct an edge-weighted graph $\mathcal{G}_\mathsf{c}$ as follows: If $\mathsf{dist}(s, s') \le R_\mathsf{c}$, we add an edge between s and s' (It is easy to see that $\mathcal{G}_\mathsf{c}$ is in fact a unit disk graph). $\mathcal{G}_\mathsf{c}$ is called *the communication graph.* Recall that MIN-CSC requires us to find a set of vertices that induces a connected subgraph in the communication graph $\mathcal{G}_\mathsf{c}$.

First, we note that $\mathcal{G}_\mathsf{c}$ may have several connected components. We can see any feasible solution must be contained in a single connected component (otherwise, the solution can not induce a connected graph). Our algorithm tries to find a solution in every connected component. Our final solution will be the one with the minimum cost among all connected component. Note that for some connected component, there may not be a feasible solution in that component (some target point can not be covered by any point in that component), and our algorithm ignores such component.

From now on, we fix a connected component $\mathcal{C}$ in $\mathcal{G}_\mathsf{c}$. Let $\mathcal{G}[\mathcal{C}]$ be the collection of all edges in the connected component $\mathcal{C}$. Similar with Wu et al. [37], we formulate the MIN-CSC problem as a group Steiner tree (GST) problem. Each edge $e \in \mathcal{G}[\mathcal{C}]$ is associated with a cost $c_e = 1$. For each target $p \in \mathcal{P}$, we create a group

$$\mathsf{gp}(p) = \mathcal{C} \cap D(p, R_\mathsf{s}) = \{s \mid s \in \mathcal{C}, \mathsf{dist}(p, s) \le R_\mathsf{s}\}.$$

The goal is to find a tree $\mathcal{T}$ (in $\mathcal{G}[\mathcal{C}]$) such that $\mathcal{T} \cap \mathsf{gp}(p) \neq \emptyset$ for all $p \in \mathcal{P}$ and the cost is minimized. We can easily see the GST instance constructed above is equivalent to the original MIN-CSC problem (the cost of the tree $\mathcal{T}$ is the number of nodes in $\mathcal{T}$ minus 1). The GST problem can be formulated as the following linear integral program: We pick a root $r \in \mathcal{C}$ for the tree $\mathcal{T}$ (we need to enumerate all possible roots). For each edge $e \in \mathcal{G}[\mathcal{C}]$, we use Boolean variable x_e to denote whether we choose edge e.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in \mathcal{G}[\mathcal{C}]} x_e \qquad (1) \\ \text{subject to} \quad & \sum_{e \in \partial(S)} x_e \ge 1, \quad \text{for all } S \subset \mathcal{C} \text{ such that } r \in S \text{ and } \exists p, S \cap G_p = \emptyset; \\ & x_e \in \{0, 1\}, \quad \forall e \in \mathcal{G}[\mathcal{C}]. \end{aligned}$$

The second constraint says that for any cut $\partial(S)$ that separates the root r from any group, there must be at least one chosen edge. By replacing $x_e \in \{0, 1\}$ with $x \in [0, 1]$, we obtain the linear programming relaxation of (1) (denoted as Lp-GST). By the duality between flow and cut, we can see that the second constraint is equivalent to dictating that we can send at least 1 unit of flow from the root r to nodes in $\mathsf{gp}(p)$, for each p. This flow viewpoint (also observed in the original GST paper [16]) will be particularly useful to us later. So we write down the flow LP explicitly as follows. We first replace every undirected edge $e = (u, v)$ by two directed arcs (u, v) and (v, u). For each $p \in \mathcal{P}$ and each directed arc (u, v), we have a variable x^p_{uv} indicating the flow of commodity p on arc (u, v). We use $y^p_v = \sum_u x^p_{uv} - \sum_w x^p_{vw}$ to denote the net flow (also called *flow excess*) of commodity p into node v. Then Lp-GST can be equivalently rewritten as the following linear program (denoted as Lp-flow):

$$\begin{aligned}
\text{minimize} \quad & \sum_{(u,v)\in\mathcal{G}[\mathcal{C}]} x_{uv} && (2)\\
\text{subject to} \quad & y_v^p = \sum_u x_{uv}^p - \sum_w x_{vw}^p \quad \text{for all } v \in \mathcal{C} \\
& y_r^p = -1 \quad \text{for all } p \in \mathcal{P}, \\
& \sum_{v\in \mathsf{gp}(p)} y_v^p \geq 1 \quad \text{for all } p \in \mathcal{P}, \\
& y_u^p = 0 \quad \text{for all } u \notin \mathsf{gp}(p), u \neq r, \\
& x_{uv}^p \leq x_{uv} \quad \text{for all } p \in \mathcal{P}, u, v \in \mathcal{C}, \\
& x_{uv}^p, y_v^p \in [0,1], \quad \text{for all } u, v \in \mathcal{G}[\mathcal{C}].
\end{aligned}$$

Now, we describe our algorithm. Our algorithm mainly consists of two steps. In the first step, we extract a *geometric hitting set* instance from the optimal fractional solution of Lp-flow. We can find an integral solution H for the hitting set problem and we can show its cost is at most $O(C^2\mathsf{OPT})$. Moreover all sensors in H can cover all target points $p \in \mathcal{P}$. In the second step, we extract a Steiner tree instance, again from the optimal fractional solution of Lp-flow. We show it is possible to round the Steiner tree LP to get a constant approximation integral Steiner tree, which can connect all points in H.

Step 1: Constructing the Hitting Set Instance :

We first solve the linear program Lp-flow and obtain the fractional optimal solution (x_{uv}, y_v). Let Opt(Lp-flow) to denote the optimal value of Lp-flow. We place a grid with grid size $l = \frac{\sqrt{2}}{2}$ in the plane (i.e., each cell is a $\frac{\sqrt{2}}{2} \times \frac{\sqrt{2}}{2}$ square). For each $p \in \mathcal{P}$, consider the set of sensors $\mathsf{gp}(p)$, that is the set of sensors which can cover p. Since $\mathsf{gp}(p)$ is contained in a disk of radius $R_\mathsf{s} \leq C$, there are at most $\frac{\sqrt{2}}{2}O(C^2) = O(1)$ grid cells that may contain some points in $\mathsf{gp}(p)$. Since $\sum_{v\in\mathsf{gp}(p)} y_v^p \geq 1$, there must be a cell (say $\mathsf{cl}(p)$) such that

$$\sum_{v\in \mathsf{gp}(p)\cap \mathsf{cl}(p)} y_v^p \geq 1/3C^2 = \Omega(1). \tag{3}$$

We call $\mathsf{cl}(p)$ the *significant cell* for point p. [3]

Now, we construct a geometric hitting set (HitSet) instance $(\mathcal{U}, \mathcal{F})$ as follows: Let the set of points be $\mathcal{U} = \cup_{p\in\mathcal{P}}(\mathsf{gp}(p) \cap \mathsf{cl}(p))$ and the family of subsets be $\mathcal{F} = \{\mathsf{gp}(p)\}_{p\in\mathcal{P}}$. The goal is to choose a subset H of $\mathcal{U}$ such that $\mathsf{gp}(p) \cap H \neq \emptyset$ for all $p \in \mathcal{P}$ (i.e., we want to hit every set in $\mathcal{F}$). Write the linear program relaxation for the HitSet problem (denoted as Lp-HS):

$$\begin{aligned}
\text{minimize} \quad & \sum_{u\in\mathcal{U}} z_u && (4)\\
\text{subject to} \quad & \sum_{u\in \mathsf{gp}(p)\cap \mathsf{cl}(p)} z_u \geq 1 \quad \text{for all } p \in \mathcal{P}, \\
& z_u \in [0,1], \quad \text{for all } u \in \mathcal{U}.
\end{aligned}$$

Let Opt(Lp-HS) to denote the optimal value of Lp-HS. We need the following simple lemma.

[3] If there are multiple such cells, we pick one arbitrarily.

Lemma 3 $\mathsf{Opt}(\mathsf{Lp\text{-}HS}) \leq 3C^2\mathsf{Opt}(\mathsf{Lp\text{-}flow})$.

Proof: Suppose (x_{uv}, y_v) is the optimal fractional solution for Lp-flow. Now, we want to construct a feasible fractional solution $\{z_u\}_{u\in\mathcal{U}}$ for Lp-HS such that $\sum_{u\in\mathcal{U}} z_u \leq O(C^2\sum_{uv} x_{uv}) = O(C^2\mathsf{Opt}(\mathsf{Lp\text{-}flow}))$. We simply let

$$z_u = \min\{1, 3C^2 \max_{p\in\mathcal{P}} y^p_u\}.$$

From (3), we can easily see z_u is a feasible solution for the HitSet problem:

$$\sum_{u\in\mathsf{gp}(p)\cap\mathsf{cl}(p)} z_u \geq \sum_{u\in\mathsf{gp}(p)\cap\mathsf{cl}(p)} \min\{1, 3C^2 y^p_u\} \geq 1 \quad \text{for all } p\in\mathcal{P}$$

It remains to see that

$$\begin{aligned}\sum_{u\in\mathcal{U}} z_u &\leq \sum_{u\in\mathcal{U}} 3C^2 \max_p y^p_u \leq 3C^2 \sum_{u\in\mathcal{U}} \max_{p\in\mathcal{P}} \left(\sum_{w\in\mathcal{C}} x^p_{wu}\right) \\ &\leq 3C^2 \sum_{u\in\mathcal{U}}\sum_{w\in\mathcal{C}} \max_{p\in\mathcal{P}} (x^p_{wu}) = 3C^2 \sum_{u\in\mathcal{U}}\sum_{w\in\mathcal{C}} x_{wu} \\ &\leq 3C^2 \sum_{uv} x_{uv}.\end{aligned}$$

This finishes the proof. □

Bronnimann et al. [2], combined with the existence of ϵ-net of size $O(1/\epsilon)$ for disks (see e.g., [31]), showed that we can round the above linear program Lp-HS to obtain an integral solution (i.e., an actual hitting set) $H \subset \mathcal{U}$ such that $|H| \leq O(\mathsf{Opt}(\mathsf{Lp\text{-}HS}))$ (the connection to ϵ-net was made simpler and more explicit in Even et al. [12]). Hence, by Lemma 3, we have that $|H| \leq O(C^2\mathsf{Opt}(\mathsf{Lp\text{-}flow}))$.

Step 2: Constructing the Steiner Tree Instance : We now have a hitting set $H \subset \mathcal{U}$. Consider a node $u \in H$. Since u is a node (a sensor) in the hitting set, we know there is some point $p_u \in \mathcal{P}$ such that $u \in \mathsf{gp}(p_u) \cap \mathsf{cl}(p_u)$. In other words, u can cover p_u and is in the significant cell of p_u. From (3), we know that $\sum_{v\in\mathsf{gp}(p_u)\cap\mathsf{cl}(p_u)} y^{p_u}_v \geq \Omega(1/C^2)$.

Consider the set of cells $\Delta = \{\mathsf{cl}(p_u) \mid u \in H\}$ [4] If there is a cell which contains the root r, we exclude it from Δ. From each cell $\mathsf{cl}(p) \in \Delta$, we pick an arbitrary node (i.e., sensor) $v(\mathsf{cl}(p))$ in it, called the *representative node* of cl. By 3 (i.e., $\sum_{v\in\mathsf{gp}(p)\cap\mathsf{cl}(p)} y^p_v \geq \Omega(1/C^2)$), at least $\Omega(1/C^2)$ flow of commodity p that enters $\mathsf{cl}(p)$.

Consider the Steiner tree problem in $\mathcal{G}(\mathcal{C})$ in which the set of terminals is defined to be

$$\mathsf{Ter} = \{r\} \cup \{v(\mathsf{cl}) \mid \mathsf{cl} \in \Delta\}.$$

In another word, the goal of this Steiner tree problems is to connect r and all representative nodes. We write down the following linear program relaxation for the Steiner tree problem (denoted as Lp-ST):

$$\begin{aligned} \text{minimize} \quad & \sum_{e\in\mathcal{G}[\mathcal{C}]} x_e \\ \text{subject to} \quad & \sum_{e\in\partial(S)} x_e \geq 1, \quad \text{for all } S\subset\mathcal{C} \text{ such that } r\in S \text{ and } \exists \mathsf{cl}\in\Delta, v(\mathsf{cl})\notin S \\ & x_e \in [0,1], \quad \forall e\in\mathcal{G}[\mathcal{C}]. \end{aligned} \tag{5}$$

[4] If a cell is the significant cell for more than one node p_u, Δ only has one copy of the cell. In other words, it is indeed a *set* of cells.

Now, we construct a feasible fractional solution for $\mathsf{Lp\text{-}ST}$ as follows. Consider the optimal fractional solution (x_{uv}, y_v) of $\mathsf{Lp\text{-}flow}$. We would like to construct another feasible fractional solution $(\widehat{x}_{uv}, \widehat{y}_v)$ for $\mathsf{Lp\text{-}flow}$. First, we construct an intermediate solution $(\widetilde{x}_{uv}, \widetilde{y}_v)$ by *rerouting* some flow. Then, we scale the flow to construct $(\widehat{x}_{uv}, \widehat{y}_v)$. The details are as follows:

- (Flow Rerouting) Consider a cell $\mathsf{cl}(p) \in \Delta$. For each node $u \in \mathsf{gp}(p) \cap \mathsf{cl}(p)$, let $\widetilde{x}^p_{uv(\mathsf{cl}(p))} \leftarrow x^p_{uv(\mathsf{cl}(p))} + y^p_u$, and let $\widetilde{x}^p_{uv} \leftarrow x^p_{uv}$ for any node $v \neq v(\mathsf{cl}(p))$. In other words, we route the flow excess at node u to node $v(\mathsf{cl}(p))$. After such updates, for each $u \in \mathsf{gp}(p) \cap \mathsf{cl}(p), u \neq v(\mathsf{cl}(p))$ we can see the flow excess is zero, or equivalently $\widetilde{y}^p_u = 0$. The flow excess at node $v(\mathsf{cl}(p))$ is

$$\widetilde{y}^p_{v(\mathsf{cl}(p))} = \sum_{v \in \mathsf{gp}(p) \cap \mathsf{cl}(p)} y^p_v \geq 1/3C^2.$$

  We repeat the above process for all $\mathsf{cl}(p) \in \Delta$.

- We next increase the flow excess at node $v(\mathsf{cl}(p))$ to 1 for all $\mathsf{cl}(p) \in \Delta$, and construct another feasible solution $(\widehat{x}_{uv}, \widehat{y}_v)$. For each $\mathsf{cl}(p) \in \Delta$, we define $(\widehat{x}^p_{uv}, \widehat{y}^p_v)$ as follows:

  1. For each edge e, let $\widehat{x}^p_e \leftarrow \widetilde{x}^p_e / \widetilde{y}^p_{v(\mathsf{cl}(p))}$. Note that such scaling increases the flow excess at node $v(\mathsf{cl}(p))$ by a $1/\widetilde{y}^p_{v(\mathsf{cl}(p))}$ factor.
  2. For each node v, let $\widehat{y}^p_v \leftarrow \widetilde{y}^p_v / \widetilde{y}^p_{v(\mathsf{cl}(p))}$.

  After the scaling, 1 unit flow (thinking $\widehat{x}^p_{uv}$ as the flow value on (u, v)) enters $v(\mathsf{cl}(p))$ and $\widehat{y}^p_{v(\mathsf{cl}(p))} = 1$. On the other hand, we have that $\widehat{x}^p_{uv} = \widetilde{x}^p_{uv} / \widetilde{y}^p_{v(\mathsf{cl}(p))} \leq 3C^2 \widetilde{x}^p_{uv}$ for each edge e following from the fact that $1/\widetilde{y}^p_{v(\mathsf{cl}(p))} \leq 3C^2$. Moreover, for each $\mathsf{cl}(p) \in \Delta$ and each node $u \in \mathsf{gp}(p) \cap \mathsf{cl}(p)$, we have

$$\widehat{x}^p_{uv(\mathsf{cl}(p))} = \widetilde{x}^p_{uv(\mathsf{cl}(p))} / \widetilde{y}^p_{v(\mathsf{cl}(p))} = (x^p_{uv(\mathsf{cl}(p))} + y^p_u) / \widetilde{y}^p_{v(\mathsf{cl}(p))} \leq 3C^2 \cdot x^p_{uv(\mathsf{cl}(p))} + y^p_u / \widetilde{y}^p_{v(\mathsf{cl}(p))}. \quad (6)$$

Here the equality follows from the definition of $\widetilde{x}^p_{uv(\mathsf{cl}(p))}$.

Let $\check{x}_e = \max_{p \in \mathcal{P}} \widehat{x}^p_{uv} + \max_{p \in \mathcal{P}} \widehat{x}^p_{vu}$, where e is the undirected edge corresponding to directed edges uv and vu (Notice that $\mathsf{Lp\text{-}flow}$ is formulated on directed graphs and Steiner tree is formulated on undirected graphs.). For each $v(\mathsf{cl}(p))$, since at least 1 unit flow (thinking $\widehat{x}^p_{uv}$ as flow value on (u, v)) enters $v(\mathsf{cl}(p))$ and $\check{x}_e \geq \widehat{x}^p_e$, $\check{x}_e$ is a feasible solution for $\mathsf{Lp\text{-}ST}$.

Next, we show the optimal value of $\mathsf{Lp\text{-}ST}$ is not much larger than that of $\mathsf{Lp\text{-}flow}$.

Lemma 4 $\mathsf{Opt}(\mathsf{Lp\text{-}ST}) \leq O(C^2 \mathsf{Opt}(\mathsf{Lp\text{-}flow}))$.

Proof: Recall $\check{x}_e$ is a feasible solution for $\mathsf{Lp\text{-}ST}$ and x_{uv} is the optimal solution for $\mathsf{Lp\text{-}flow}$. Also recall that $H \subset \mathcal{U}$ is a hitting set instance satisfying that $|H| \leq O(C^2 \mathsf{Opt}(\mathsf{Lp\text{-}flow}))$. We only need to show that

$$\sum_{e \in \mathcal{G}[\mathcal{C}]} \check{x}_e \leq O(C^2 \sum_{(u,v) \in \mathcal{G}[\mathcal{C}]} x_{uv} + |H|).$$

This can be seen as follows:

$$
\begin{aligned}
\sum_{e\in\mathcal{G}[\mathcal{C}]} \check{x}_e &= \sum_{(u,v)\in\mathcal{G}[\mathcal{C}]} \max_{p\in\mathcal{P}} \widehat{x}^p_{uv} = \sum_{(u,v)\in\mathcal{G}[\mathcal{C}]} \max_{p\in\mathcal{P}} \widetilde{x}^p_{uv}/\widetilde{y}^p_{v(\mathsf{cl}(p))} \\
&\le \sum_{(u,v)\in\mathcal{G}[\mathcal{C}]} \max_{p\in\mathcal{P}} x^p_{uv}/\widetilde{y}^p_{v(\mathsf{cl}(p))} + \sum_{\mathsf{cl}(p)\in\Delta, u\in\mathsf{gp}(p)\cap\mathsf{cl}(p)} y^p_u/\widetilde{y}^p_{v(\mathsf{cl}(p))} \\
&\le 3C^2 \sum_{(u,v)\in\mathcal{G}[\mathcal{C}]} \max_{p\in\mathcal{P}} x^p_{uv} + \sum_{v(\mathsf{cl}(p))} 1 \\
&\le 3C^2 \sum_{(u,v)\in\mathcal{G}[\mathcal{C}]} x_{uv} + |H|.
\end{aligned}
$$

The second equality follows from the construction of $\widehat{x}^p_{uv}$ and (6). The first inequality follows from the definition of $\widetilde{x}^p_{uv}$ (we only reroute the flow for commodity p such that $\mathsf{cl}(p) \in \Delta$, hence the second term). The second inequality follows from the fact that $\widetilde{y}^p_{v(\mathsf{cl}(p))} \ge 1/3C^2$. This finishes the proof of the lemma. □

It is well known that the integrality gap of the Steiner tree problem is a constant [35]. In particular, it is known that using the primal-dual method (based on Lp-ST) in [17] (see also [35, Chapter 7.2]), we can obtain an integral solution $\overline{x}_e$ such that

$$\sum_{e\in\mathcal{G}[\mathcal{C}]} \overline{x}_e \le 2\mathsf{Opt}(\mathsf{Lp\text{-}ST}) \le O(C^2\mathsf{Opt}(\mathsf{Lp\text{-}flow})) \le O(C^2\mathsf{OPT}).$$

Let J be the set of vertices spanned by the integral Steiner tree $\{\overline{x}_e\}$. The above discussion shows that $|J| \le O(C^2\mathsf{OPT})$. Our final solution (the set of sensors we choose) is $\mathsf{Sol} = H \cup J$. The feasibility of Sol is proved in the following simple lemma.

Lemma 5 Sol *is a feasible solution.*

Proof: We only need to show that Sol induces a connected graph and covers all the target points. Obviously, H covers all target points, so does Sol. Since J is a Steiner tree, thus connected. Moreover, J connects all representatives $v(\mathsf{cl})$ for all $\mathsf{cl} \in \Delta$. On the other hand, H ony contains those sensors in $\mathsf{cl} \in \Delta$. So every sensor in $v \in H$ (say $v \in \mathsf{cl}$) is connected to the representative $v(\mathsf{cl})$. So $H \cup J$ induces a connected subgraph. □

Lastly, we need to show the performance guarantee. This is easy since we have shown that both $|H| \le O(C^2\mathsf{OPT})$ and $|J| \le O(C^2\mathsf{OPT})$. So $|\mathsf{Sol}| = O(C^2\mathsf{OPT}) = O(\mathsf{OPT})$ since C is assumed to be a constant.

4 Budgeted Connected Sensor Cover

Again we assume that $R_\mathsf{c} = 1$ and $R_\mathsf{s} = C$. Recall that our goal is to find a subset $\mathcal{S}' \subseteq \mathcal{S}$ of sensors with cardinality B which induces a connected subgraph and covers as many targets as possible. We first construct the communication graph $\mathcal{G}_\mathsf{c}$ as in Section 3. Again, we only need to focus on a connected component of $\mathcal{G}_\mathsf{c}$. Then we find a square Q in the Euclidean plane large enough such that all of the n sensors are inside Q. We partition Q into small square cells of equal size. Let the side length of each cell be $l = \frac{\sqrt{2}}{2}$. Denote the cell in the ith row and jth column of the partition as $\mathsf{cl}_{i,j}$. Let $V_{i,j} = \{v \in \mathcal{S} \mid v \in \mathsf{cl}_{i,j}\}$ be the collection of sensors in $\mathsf{cl}_{i,j}$. We then partition these cells into k^2 different cell groups $\mathsf{CG}_{a,b}$, where $k = \lceil 2C/l + 1 \rceil$. In particular, we let

$$\mathsf{CG}_{a,b} = \{\mathsf{cl}_{i,j} \mid i \equiv a (\bmod k), j \equiv b (\bmod k)\} \text{ for } a \in [k], b \in [k].$$

Algorithm 1: Reassign profits via the greedy algorithm

Input: The sensor collection $\mathcal{S}$, the target collection $\mathcal{P}$, the cell collection $\mathsf{CG}_{a,b}$.
Output: Profit function $\widehat{p} : \mathcal{P} \to \mathbb{Z}^+ \cup \{0\}$

1. **for all** $\mathsf{cl}_{i,j} \in \mathsf{CG}_{a,b}$ **do**
2. $P_t \leftarrow \mathcal{P}$ //P_t is the set of uncovered targets
3. $V_s \leftarrow V_{i,j}$ //V_s is the set of available sensors
4. **for all** $v \in \mathcal{S}$ **do**
   (a) $v \leftarrow \arg\max_{v \in P_t} |N_{P_t}(v)|$ //$N_{P_t}(v)$ is the set of uncovered targets that can be covered by v.
   (b) $\widehat{p}(v) \leftarrow |N_{P_t}(v)|$, $P_t \leftarrow P_t \backslash N_{P_t}(v)$, $V_s \leftarrow V_s \backslash \{v\}$
5. **end for**
6. **end for**
7. return $\widehat{p}$

and $\mathcal{V}_{a,b} = \mathcal{S} \cap \mathsf{CG}_{a,b}$ be the collection of sensors in $\mathsf{CG}_{a,b}$.

With the above value k, we make a simple but useful observation as follows.

Observation 1 *There is no target covered by two different sensors contained in two different cells of* $\mathsf{CG}_{a,b}$.

Denote the optimal solution of Budgeted-CSC problem as OPT. In this section, we present an $O\big(\frac{1}{C^2}\big)$ factor approximation algorithm for the Budgeted-CSC problem.

4.1 The Algorithm

For $0 \le a, b < k$, we repeat the following two steps, and output a tree T with $O\big(B\big)$ vertices (sensors) which covers the maximum number of targets. Then based on T, we find a subtree $\tilde{T}$ with exactly B vertices as our final output.

Step 1: Reassign profit : The profit $p(S)$ of a subset $S \subseteq \mathcal{S}$ is the number of targets covered by S. $p(S)$ is a submodular function. In this step, we design a new profit function (called *modified profit function*) $\widehat{p} : \mathcal{S} \to \mathbb{Z}^+$ for the set of sensors. To some extent, $\widehat{p}$ is a linearized version of p (module a constant approximation factor).

Now, we explain in details how $\widehat{p}$ is defined. Fix a cell group $\mathsf{CG}_{a,b}$. [5] For the vertices in $\mathcal{V}_{a,b}$, we use the greedy algorithm Algorithm 1 to reassign profits of the vertices in $\mathcal{V}_{a,b}$. Generally speaking, we greedily pick a vertex which covers the most number of targets each time, and use this number as the modified profit. The details are as follows. Among all vertices in $\mathcal{V}_{a,b}$, we pick a vertex v_1 which can cover the most number of targets, and use this number as its modified profit $\widehat{p}(v_1)$. Remove the chosen vertex and targets covered by it. We continue to pick the vertex v_2 in $\mathcal{V}_{a,b}$ which can cover the most number of uncovered targets. Set the modified profit $\widehat{p}(v_2)$ to be the number of newly covered targets. Repeat the above steps until all the sensors in $\mathcal{V}_{a,b}$ have been picked out. For other vertices v which are not in $\mathcal{V}_{a,b}$, we simply set their modified profit $\widehat{p}(v)$ as 0.

[5] For each $\mathsf{CG}_{a,b}$, we define a modified profit function $\widehat{p}_{a,b}$. For ease of notation, we omit the subscripts.

Let us first make some simple observations about p and $\widehat{p}$. We use $\widehat{p}(S)$ to denote $\sum_{v\in S}\widehat{p}(v)$. First, it is not difficult to see that $\widehat{p}(S) \le p(S)$ for any subset $S \subseteq \mathcal{S}$. Second, we can see that it is equivalent to run the greedy algorithm for each cell in $\mathsf{CG}_{a,b}$ separately (due to Observation 1). Suppose $S_1 \subseteq \mathsf{cl}_{c,d}$, $S_2 \subseteq \mathsf{cl}_{c',d'}$ where $\mathsf{cl}_{c,d}$ and $\mathsf{cl}_{c',d'}$ are two different cells in $\mathsf{CG}_{a,b}$, then $p(S_1 \cup S_2) = p(S_1) + p(S_2)$ due to Observation 1.

Consider a cell $\mathsf{cl}_{c,d} \in \mathsf{CG}_{a,b}$. Let $D_{c,d} = \{v_1, v_2, ..., v_n\} \subseteq \mathsf{cl}_{c,d} \cap \mathcal{S}$, where the vertices are indexed by the order in which they were selected by the greedy algorithm. Let $D^i_{c,d} = \{v_1, v_2, ..., v_i\}$ be the first i vertices in $D_{c,d}$. By the following lemma, we can see that the modified profit function $\widehat{p}$ is a constant approximation to true profit function p over any vertex subset $V \subseteq \mathcal{V}_{a,b}$.

Lemma 6 *For a set of vertices V in the same cell $\mathsf{cl}_{c,d} \in \mathsf{CG}_{a,b}$, such that $|V| \le i$, we have that $p(D^i_{c,d}) = \widehat{p}(D^i_{c,d}) \ge (1 - 1/e)p(V)$.*

Proof: By the greedy rule, we can see $p(D^i_{c,d}) = \widehat{p}(D^i_{c,d})$. By Lemma 1, we know that $\widehat{p}(D^i_{c,d}) \ge (1 - 1/e)\max_{|V|\le i} p(V)$. □

Step 2: Guess the optimal profit and calculate a tree T : Although the actual profit of OPT is unknown, we can guess the profit of OPT (by enumerating all possibilities). For each $0 \le a, b < k$, we calculate in this step a tree T of size at most $4B$, using the QST algorithm (see Lemma 2). We can show that among these trees (for different a, b values), there must be one tree of profit no less than $\frac{1}{k^2}\left(1 - \frac{1}{e}\right)\mathsf{OPT}$.

After choosing the best tree T with the highest profit, we construct a subtree $\tilde{T}$ of size B based on T as our final solution of Budgeted-CSC.

Algorithm 2: Algorithm for Budgeted-CSC with greedy profit assignment

---

Input: The sensor collection $\mathcal{S}$, the target collection $\mathcal{P}$, budget B.

Output: a tree $\tilde{T}$ with $|\tilde{T}| \le B$.

1. Construct the communication graph $\mathcal{G}_{\mathsf{c}}$
2. **for** a **from** 0 to $k-1$, b **from** 0 to $k-1$
   - (a) Reassign every vertex's profit with Algorithm 1
   - (b) Set every edge's cost as 1
   - (c) $ProfitOpt_{guess} \leftarrow 1$
   - (d) **Do**
     - i. $T' \leftarrow$ Run the 2-approximation algorithm of QST on $\mathcal{G}_{\mathsf{c}}$ with quota $ProfitOpt_{guess}$
     - ii. **if** $|T'| \le 4B$ **then** $T \leftarrow T'$
     - iii. $ProfitOpt_{guess} = ProfitOpt_{guess} + 1$
   - (e) **While**$(|T'| \le 4B)$
3. **end for**
4. $\tilde{T} \leftarrow$ use the dynamic programming algorithm described in Section 5.2.2 in [24] to find the best profit subtree of size B from T.
5. return $\tilde{T}$

We first show that there exists $0 \leq a, b < k$, such that based on the modified profit $\widehat{p}$ on $\mathsf{CG}_{a,b}$, there exists a tree with at most $2B$ vertices of total modified profit at least $\frac{1}{k^2}\left(1-\frac{1}{e}\right)\mathsf{OPT}$. We use T_{OPT} to denote the set of vertices of the optimal solution.

Lemma 7 *There exists a tree T_0 in $\mathcal{G}_{\mathsf{c}}$, $|T_0| \leq 2B$ such that $\widehat{p}(T_0) \geq \dfrac{1}{k^2}\left(1-\dfrac{1}{e}\right)\mathsf{OPT}$*

Proof: We first notice that

$$\mathsf{OPT} = p\left(\bigcup_{0\leq a,b<k} T_{\mathsf{OPT}} \cap \mathsf{CG}_{a,b}\right) \leq \sum_{0\leq a,b<k} p\left(T_{\mathsf{OPT}} \cap \mathsf{CG}_{a,b}\right).$$

Hence, there exists $0 \leq a', b' < k$, such that

$$p(T_{\mathsf{OPT}} \cap \mathsf{CG}_{a',b'}) \geq \frac{1}{k^2}\sum_{0\leq a,b<k} p(T_{\mathsf{OPT}} \cap \mathsf{CG}_{a,b}) \geq \frac{1}{k^2}\mathsf{OPT}.$$

For any cell $\mathsf{cl}_{c,d} \in \mathsf{CG}_{a',b'}$, suppose $n_{c,d} = |T_{\mathsf{OPT}} \cap \mathsf{cl}_{c,d}|$. T_0 is obtained from T_{OPT} by appending all vertices in $D_{c,d}^{n_{c,d}}$ (recall that $D_{c,d}^{n_{c,d}}$ consists of the first $n_{c,d}$ vertices selected in $\mathsf{cl}_{c,d}$ by the greedy algorithm). Note that we append at most B vertices in total, and all vertices are still connected (since all vertices in the same cell are connected). Thus, T_0 is connected and has at most $2B$ vertices.

By Lemma 6, we can see that $\widehat{p}(D_{c,d}^{n_{c,d}}) \geq \left(1-\frac{1}{e}\right)p(T_{\mathsf{OPT}} \cap \mathsf{cl}_{c,d})$. Thus, we have

$$\begin{aligned}\widehat{p}(T_0) &= \sum_{\mathsf{cl}_{c,d}\in\mathsf{CG}_{a,b}} \widehat{p}(D_{c,d}^{n_{c,d}}) \geq \left(1-\frac{1}{e}\right)\sum_{\mathsf{cl}_{c,d}\in\mathsf{CG}_{a,b}} p\left(T_{\mathsf{OPT}} \cap \mathsf{cl}_{c,d}\right) \\ &= \left(1-\frac{1}{e}\right)p(T_{\mathsf{OPT}} \cap \mathsf{CG}_{a,b}) \geq \frac{1}{k^2}\left(1-\frac{1}{e}\right)\mathsf{OPT}.\end{aligned}$$

The second equality holds due to Observation 1. □

Then, by Lemma 2 and Lemma 7, if we run the QST algorithm (with $\widehat{p}$ as the profit function), we can obtain the suitable tree T with at most $4B$ vertices of profit at least $\frac{1}{k^2}\left(1-\frac{1}{e}\right)p(\mathsf{OPT})$. The pseudocode of the algorithm can be found in Algorithm 2.

Lemma 8 *Let T be the tree obtained in Algorithm 2, then $p(T) \geq \frac{1}{k^2}\left(1-\frac{1}{e}\right)\mathsf{OPT}$*

Proof: By Lemma 7, we can obtain a tree T with at most $4B$. We also have $\widehat{p}(T) \geq \frac{1}{k^2}\left(1-\frac{1}{e}\right)\mathsf{OPT}$. Since $p(S) \geq \widehat{p}(S)$ for any S, we have that $p(T) \geq \frac{1}{k^2}\left(1-\frac{1}{e}\right)\mathsf{OPT}$. □

Then we show how to construct a subtree $\tilde{T}$ of B vertices based on tree T. Our technique is the same as Khuller et al. [24]. Firstly, they use the following theorem by Jordan [23] to prove Lemma 10. Then by a carefully partition, they obtain a subtree with B vertices of profit at least $\frac{1}{13}$ of original tree with $6B$ vertices. Our construction is almost the same except that the original tree T in our setting has at most $4B$ vertices.

Lemma 9 (Jordan [23]) *Given any tree on n vertices, we can decompose it into two trees (by replicating a single vertex) such that the smaller tree has at most $\lceil \frac{n}{2} \rceil$ nodes and the larger tree has at most $\lceil \frac{2n}{3} \rceil$ nodes.*

Lemma 10 (Khuller et al. [24]) *Let B be greater than a sufficiently large constant. Given a tree T with $6B$ nodes, we can decompose it into 13 trees of size at most B nodes each.*

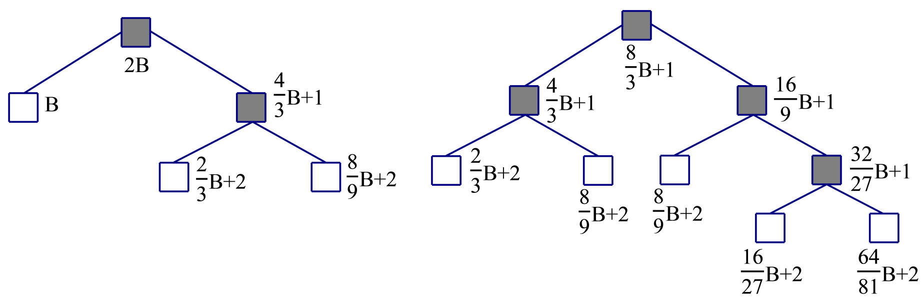
Denote the subtree with highest total profit as $\tilde{T}$, we show the following lemma.

Lemma 11 $p(\tilde{T}) \geq \frac{1}{8}p(T)$

Proof: By Lemma 9, we decompose the tree T into two trees T_1 and T_2 such that $|T_1| \leq 2B$ and $|T_2| \leq \frac{8}{3}B + 1$ and continue decomposing until the tree has at most k vertices (as shown in the figure. Note that each subtree in the white square in the figure has at most B vertices). Thus we can decompose a tree of size $4B$ to at most 8 subtrees of size at most B. See the figure. Suppose the subtrees are T_1,T_2,...,T_8. Then we have,

$$p(\tilde{T}) \geq \frac{1}{8}\sum_{i=1}^{8} p(T_i) \geq \frac{1}{8}p(T)$$

So there is a subtree of size at most k and profit at least $\frac{1}{8}p(T)$. □



Use the same dynamic programming algorithm in Khuller et al. [24], we can find $\tilde{T}$ from tree T. Combining Lemma 8 and Lemma 11, $p(\tilde{T}) \geq \frac{1}{8}\left(1-\frac{1}{e}\right)\frac{1}{(2\sqrt{2}C+1)^2}\mathsf{OPT} = \frac{1}{12.66(8C^2+4\sqrt{2}C+1)}\mathsf{OPT} \geq \frac{1}{102C^2}\mathsf{OPT}$ (if $C \geq 100$).

Thus, we have obtained Theorem 2.

5 Conclusion and Future Work

There are several interesting future directions. The first obvious open question is that whether we can get constant approximations for MIN-CSC and Budgeted-CSC without Assumption 1 (it would be also interesting to obtain approximation ratios that have better dependency on C). Generalizing the problem further, an interesting future direction is the case where different sensors have different transmission ranges and sensing ranges. Whether the problems admit better approximation ratios than the (more general) graph theoretic counterparts is still wide open.

6 Acknowledgments

We would like to thank anonymous reviewers for their constructive comments, and pointing out a problematic argument in a previous version of the paper. We also would like thank Dingzhu Du and Zhao Zhang for helpful discussions. Research supported in part by the National Basic Research Program of China Grant 2015CB358700, 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61202009, 61033001, 61361136003.

References

[1] Christoph Ambühl, Thomas Erlebach, Matúš Mihalák, and Marc Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 3–14. Springer, 2006.

[2] Hervé Brönnimann and Michael T Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(1):463–479, 1995.

[3] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[4] Gruia Calinescu and Alexander Zelikovsky. The polymatroid steiner problems. *Journal of Combinatorial Optimization*, 9(3):281–294, 2005.

[5] Timothy M. Chan, Elyot Grant, Jochen Könemann, and Malcolm Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1576–1585. SIAM, 2012.

[6] Chandra Chekuri, Guy Even, and Guy Kortsarz. A greedy approximation algorithm for the group steiner problem. *Discrete Applied Mathematics*, 154(1):15–34, 2006.

[7] Xiuzhen Cheng, Xiao Huang, Deying Li, Weili Wu, and Ding-Zhu Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[8] Kenneth L. Clarkson and Kasturi Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete & Computational Geometry*, 37(1):43–58, 2007.

[9] Decheng Dai and Changyuan Yu. A 5+ ϵ-approximation algorithm for minimum weighted dominating set in unit disk graph. *Theoretical Computer Science*, 410(8):756–765, 2009.

[10] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 624–633. ACM, 2014.

[11] Ding-Zhu Du and Peng-Jun Wan. *Connected Dominating Set: Theory and Applications*, volume 77. Springer Science & Business Media, 2012.

[12] Guy Even, Dror Rawitz, and Shimon Moni Shahar. Hitting sets when the vc-dimension is small. *Information Processing Letters*, 95(2):358–362, 2005.

[13] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 448–455. ACM, 2003.

[14] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[15] Naveen Garg. 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*, pages 396–402. ACM, 2005.

[16] Naveen Garg, Goran Konjevod, and R Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 253–259. Society for Industrial and Applied Mathematics, 1998.

[17] Michel X Goemans and David P Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[18] Sudipto Guha and Samir Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Information and computation*, 150(1):57–74, 1999.

[19] Himanshu Gupta, Zongheng Zhou, Samir R Das, and Quinyi Gu. Connected sensor cover: self-organization of sensor networks for efficient query execution. *Networking, IEEE/ACM Transactions on*, 14(1):55–67, 2006.

[20] Dorit S Hochbaum and Anu Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics*, 45(6):615–627, 1998.

[21] Yaochun Huang, Xiaofeng Gao, Zhao Zhang, and Weili Wu. A better constant-factor approximation for weighted dominating set in unit disk graph. *Journal of Combinatorial Optimization*, 18(2):179–194, 2009.

[22] David S Johnson, Maria Minkoff, and Steven Phillips. The prize collecting steiner tree problem: theory and practice. In *SODA*, volume 1, page 4. Citeseer, 2000.

[23] Camille Jordan. Sur les assemblages de lignes. *J. Reine Angew. Math*, 70(185):81, 1869.

[24] Samir Khuller, Manish Purohit, and Kanthi K Sarpatwar. Analyzing the optimal neighborhood: algorithms for budgeted and partial connected dominating set problems. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1702–1713. SIAM, 2014.

[25] Philip Klein and R Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *Journal of Algorithms*, 19(1):104–115, 1995.

[26] Tung-Wei Kuo, KC-J Lin, and Ming-Jer Tsai. Maximizing submodular set function with connectivity constraint: Theory and application to networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 1977–1985. IEEE, 2013.

[27] Jian Li and Yifei Jin. A ptas for the weighted unit disk cover problem. *arXiv preprint arXiv:1502.04918*, 2015.

[28] David Lichtenstein. Planar formulae and their uses. *SIAM journal on computing*, 11(2):329–343, 1982.

[29] Nabil Hassan Mustafa and Saurabh Ray. Ptas for geometric hitting set problems via local search. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 17–22. ACM, 2009.

[30] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[31] Evangelia Pyrga and Saurabh Ray. New existence proofs ε-nets. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 199–207. ACM, 2008.

[32] Kasturi Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 641–648. ACM, 2010.

[33] Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792. ACM, 2011.

[34] Peng-Jun Wan, Khaled M Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *INFOCOM 2002. Twenty-First annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, volume 3, pages 1597–1604. IEEE, 2002.

[35] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

[36] JK Willson, L Ding, W Wu, L Wu, Z Lu, and W Lee. A better constant-approximation for coverage problem in wireless sensor networks. *preprint*.

[37] Lidong Wu, Hongwei Du, Weili Wu, Deying Li, Jing Lv, and Wonjun Lee. Approximations for minimum connected sensor covereq:cellsum. In *INFOCOM, 2013 Proceedings IEEE*, pages 1187–1194. IEEE, 2013.

[38] Wei Zhang, Weili Wu, Wonjun Lee, and Ding-Zhu Du. Complexity and approximation of the connected set-cover problem. *Journal of Global Optimization*, 53(3):563–572, 2012.

[39] Feng Zou, Yuexuan Wang, Xiao-Hua Xu, Xianyue Li, Hongwei Du, Pengjun Wan, and Weili Wu. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science*, 412(3):198–208, 2011.