The Set Cover with Pairs Problem*

Refael Hassin and Danny Segev

School of Mathematical Sciences,
Tel-Aviv University, Tel-Aviv 69978, Israel
{hassin,segevd}@post.tau.ac.il

Abstract. We consider a generalization of the set cover problem, in which elements are covered by *pairs of objects*, and we are required to find a minimum cost subset of objects that induces a collection of pairs covering all elements. Formally, let U be a ground set of elements and let $\mathcal{S}$ be a set of objects, where each object i has a non-negative cost w_i. For every $\{i, j\} \subseteq \mathcal{S}$, let $\mathcal{C}(i, j)$ be the collection of elements in U covered by the pair $\{i, j\}$. The *set cover with pairs* problem asks to find a subset $A \subseteq \mathcal{S}$ such that $\bigcup_{\{i,j\}\subseteq A} \mathcal{C}(i, j) = U$ and such that $\sum_{i\in A} w_i$ is minimized.

In addition to studying this general problem, we are also concerned with developing polynomial time approximation algorithms for interesting special cases. The problems we consider in this framework arise in the context of domination in metric spaces and separation of point sets.

1 Introduction

Given a ground set U and a collection $\mathcal{S}$ of subsets of U, where each subset is associated with a non-negative cost, the *set cover* problem asks to find a minimum cost subcollection of $\mathcal{S}$ that covers all elements. An equivalent formulation is obtained by introducing a *covering function* $\mathcal{C} : \mathcal{S} \to 2^U$, that specifies for each member of $\mathcal{S}$ the subset of U it covers. Set cover now becomes the problem of finding a subset $A \subseteq \mathcal{S}$ of minimum cost such that $\bigcup_{i\in A} \mathcal{C}(i) = U$.

We consider a generalization of this problem, in which the covering function $\mathcal{C}$ is defined for pairs of members of $\mathcal{S}$, rather than for single members. Formally, let $U = \{e_1, \ldots, e_n\}$ be a ground set of elements and let $\mathcal{S} = \{1, \ldots, m\}$ be a set of *objects*, where each object $i \in \mathcal{S}$ has a non-negative cost w_i. For every $\{i, j\} \subseteq \mathcal{S}$, let $\mathcal{C}(i, j)$ be the collection of elements in U covered by the pair $\{i, j\}$. The objective of the *set cover with pairs* problem (SCP) is to find a subset $A \subseteq \mathcal{S}$ such that $\mathcal{C}(A) = \bigcup_{\{i,j\}\subseteq A} \mathcal{C}(i, j) = U$ and such that $w(A) = \sum_{i\in A} w_i$ is minimized. We refer to the special case in which each object has a unit weight as the *cardinality SCP* problem.

SCP is indeed a generalization of the set cover problem. A set cover instance with $U = \{e_1, \ldots, e_n\}$ and $S_1, \ldots, S_m \subseteq U$ can be interpreted as an SCP instance by defining $\mathcal{C}(i, j) = S_i \cup S_j$ for every $i \neq j$. Therefore, hardness results regarding set cover extend to SCP, and in particular the latter problem cannot be approximated within a ratio of $(1 - \epsilon) \ln n$ for any $\epsilon > 0$, unless NP $\subset$ TIME$(n^{O(\log \log n)})$ [4].

* Due to space limitations, most proofs are omitted from this extended abstract. We refer the reader to the full version of this paper [8], in which all missing proofs are provided.

R. Ramanujam and S. Sen (Eds.): FSTTCS 2005, LNCS 3821, pp. 164–176, 2005.

1.1 Applications

In addition to studying the SCP problem, we are concerned with developing polynomial time approximation algorithms for interesting special cases, that arise in the context of domination in metric spaces and separation of point sets.

Remote Dominating Set. Let $M = (V, d)$ be a finite metric space, and let $r_1 \leq r_2$ be two *covering radii*. We refer to the elements of $V = \{v_1, \dots, v_n\}$ as points or vertices, and assume that each $v \in V$ is associated with a non-negative cost c_v. A subset of points $S \subseteq V$ is called a *remote dominating set* if for every $v \in V$ there is a point $u \in S$ within distance r_1 of v, or a pair of points $u_1 \neq u_2 \in S$ within distance r_2 of v each. The remote dominating set problem (RDS) asks to find a minimum cost remote dominating set in M.

RDS can be interpreted as a special case of SCP: The set of elements to cover is V, which is also the set of covering objects, and the collection of points covered by $u \neq v \in V$ is

$$\mathcal{C}(u, v) = \{w \in V : \min\{d(u, w), d(v, w)\} \leq r_1 \text{ or } \max\{d(u, w), d(v, w)\} \leq r_2\} \ .$$

When $d(u, v) \in \{1, 2\}$ for every $u \neq v$ and $r_1 = r_2 = 1$, RDS reduces to the standard *dominating set* problem. Therefore, hardness results regarding set cover extend to RDS, as the dominating set problem is equivalent to set cover with regard to inapproximability.

We also consider two special cases of this problem, for which significantly better approximation algorithms are possible. In the *cardinality RDS on a tree* problem, the metric d is generated by a tree $T = (V, E)$ with unit length edges, and the covering radii are $r_1 = 1$ and $r_2 = 2$. In the *cardinality Euclidean RDS* problem, V is a set of points in the plane, and $d(u, v) = \|u - v\|_2$.

Group Cut on a Path. Let $P = (V, E)$ be a path, in which each edge $e \in E$ has a non-negative cost c_e, and let $G_1, \dots, G_k$ be k *groups*, where each group is a set of at least two vertices. A group G_i is *separated by the set of edges* $F \subseteq E$ if there is a representative $v_i \in G_i$ such that no vertex in $G_i \setminus \{v_i\}$ belongs to the connected component of $P - F$ that contains v_i. The objective of the group cut on a path problem (GCP) is to find a minimum cost set of edges that separates all groups.

Given a GCP instance we may assume without loss of generality that any optimal solution contains at least two edges. This assumption implies that GCP is a special case of SCP: The elements to cover are the groups $G_1, \dots, G_k$, and the covering objects are the edges. The groups covered by pairs of edges are defined as follows. Let $v_1, \dots, v_r$ be the left-to-right order of the vertices in G_i, and let $[v_i, v_j]$ be the set of edges on the subpath connecting v_i and v_j. The group G_i is covered by a pair of edges $e' \neq e'' \in E$ if $\{e', e''\} \cap ([v_1, v_2] \cup [v_{r-1}, v_r]) \neq \emptyset$ or if $e' \in [v_{t-1}, v_t]$ and $e'' \in [v_t, v_{t+1}]$ for some $2 \leq t \leq r - 1$.

1.2 Our Results

In Section 2 we study a natural extension of the greedy set cover algorithm [1,11,13] to approximate SCP. We define a class of functions, called *feasible maps*, that assign the elements in U to pairs of objects in the optimal solution, and characterize them by *max* and *mean* properties. We then present a conditional analysis of the greedy algorithm, based on the existence of such maps. Specifically, we prove an approximation guarantee of αH_n for the weighted and cardinality versions of the problem, given the existence of feasible maps whose max and mean are at most α, respectively. We also prove that the unconditional approximation ratio for cardinality SCP is $O(\sqrt{n \log n})$.

We continue the discussion with indications for the hardness of SCP. First, although the set cover problem becomes trivial when each subset contains a single element, we show that the corresponding special case of SCP, where each pair of objects covers at most one element, is at least as hard to approximate as set cover. Second, the analysis of the greedy set cover algorithm in [13] shows that the integrality gap of the natural LP-relaxation of set cover is $O(\log n)$. However, we demonstrate that this property does not extend to SCP, for which the integrality gap is $\Omega(n)$.

As a first attempt at attacking the RDS problem, one might consider using the greedy SCP algorithm. However, we show in Section 3 that the approximation guarantee of this algorithm is $\Omega(\sqrt{n})$, mainly due to the observation that there are instances of RDS in which non-trivial feasible maps do not exist. Nevertheless, we provide a $2H_n$-approximation algorithm that constructs a remote dominating set by approximating two dependent set cover problems.

In Section 4 we proceed to the cardinality RDS problem on a tree $T = (V, E)$. Although this problem can be solved to optimality in $O(|V|^3)$ time using dynamic programming techniques [8], we demonstrate that it can be well approximated much faster. We first show how to map a subset of "problematic" vertices of T to a small collection of pairs in the optimal solution. We then exploit the special structure of this map to present a linear time 2-approximation algorithm, and illustrate that in general graphs this algorithm does not guarantee a non-trivial approximation ratio.

In Section 5 we present a polynomial time approximation scheme for the Euclidean RDS problem. Although we follow the general framework of Hochbaum and Maass for covering and packing problems in Euclidean spaces [10], our analysis is more involved. This is due to the use of two covering radii and the restriction that the set of points we choose must be a subset of V, instead of any set of points in the plane.

Finally, in Section 6 we discuss the hardness of approximating GCP, and in particular prove that this problem is as hard to approximate as set cover. Moreover, we identify the exact point at which GCP becomes NP-hard, by showing that this problem is polynomial time solvable when the cardinality of each group is at most 3, but as hard to approximate as vertex cover when the bound on cardinality is 4. On the positive side, we prove the existence of a feasible map whose max is at most 2. This result enables us to show that the approximation ratio of the greedy SCP algorithm for this special case is $2H_k$, where k is the number of groups to be separated.

2 Set Cover with Pairs

In this section we suggest a natural extension of the greedy set cover algorithm to approximate SCP, and present a conditional analysis based on the existence of a mapping of the elements in U to pairs of objects in the optimal solution, that satisfies certain properties. We then make use of these results to prove an approximation ratio of $O(\sqrt{n \log n})$ for cardinality SCP. We also prove that the special case in which each pair of objects covers at most one element is at least as hard to approximate as set cover, and demonstrate that the integrality gap of the natural LP-relaxation of SCP is $\Omega(n)$.

2.1 A Greedy Algorithm

The greedy SCP algorithm iteratively picks the most cost-effective object or pair of objects until all elements are covered, where cost-effectiveness is defined as the ratio between the objects costs and the number of newly covered elements. Let GR be the set of objects already picked when an iteration begins, where initially $\mathrm{GR} = \emptyset$. We define:

1. For every $i \in \mathcal{S} \setminus \mathrm{GR}$, the current covering ratio of i is
$$\frac{w_i}{|\mathcal{C}(\mathrm{GR} \cup \{i\})| - |\mathcal{C}(\mathrm{GR})|} \ .$$
2. For every $i \neq j \in \mathcal{S} \setminus \mathrm{GR}$, the current covering ratio of $\{i, j\}$ is
$$\frac{w_i + w_j}{|\mathcal{C}(\mathrm{GR} \cup \{i, j\})| - |\mathcal{C}(\mathrm{GR})|} \ .$$

In each iteration we augment GR by adding a single object $i \in \mathcal{S} \setminus \mathrm{GR}$ or a pair of objects $i \neq j \in \mathcal{S} \setminus \mathrm{GR}$, whichever attains the minimum covering ratio. The algorithm terminates when U is completely covered.

2.2 Conditional Analysis

Let $F \subseteq \mathcal{S}$ be a feasible solution, that is, a set of objects that covers the elements of U, and let $P(F) = \{\{i, j\} \subseteq F : i \neq j\}$. A function $\mathcal{M} : U \to P(F)$ is a *feasible map with respect to* F if the pair of objects $\mathcal{M}(e)$ covers e, for every $e \in U$. Given a feasible map $\mathcal{M}$, for every $\{i, j\} \in P(F)$ we use $\mathbb{I}_{\mathcal{M}}(i, j)$ to indicate whether at least one element is mapped to $\{i, j\}$. We define:

$$\max(\mathcal{M}, F) = \max_{i \in F} \sum_{j \neq i} \mathbb{I}_{\mathcal{M}}(i, j) \ , \quad \mathrm{mean}(\mathcal{M}, F) = \frac{1}{|F|} \sum_{i \in F} \sum_{j \neq i} \mathbb{I}_{\mathcal{M}}(i, j) \ .$$

In other words, $\max(\mathcal{M}, F) \leq \alpha$ if each object $i \in F$ belongs to at most α pairs to which elements are mapped. Similarly, $\mathrm{mean}(\mathcal{M}, F) \leq \alpha$ if the average number of pairs, to which elements are mapped, an object belongs to is at most α. Clearly, $\mathrm{mean}(\mathcal{M}, F) \leq \max(\mathcal{M}, F)$.

In Lemma 1 we show that given the existence of a feasible map $\mathcal{M}$ with respect to an optimal solution OPT, for which $\max(\mathcal{M}, \mathrm{OPT}) \leq \alpha$, the greedy SCP algorithm constructs a solution whose cost is within factor αH_n of optimum. In Lemma 2 we show that to obtain an approximation guarantee of αH_n for cardinality SCP, the weaker condition of $\mathrm{mean}(\mathcal{M}, \mathrm{OPT}) \leq \alpha$ is sufficient.

Lemma 1. *If there exists an optimal solution* OPT *and a feasible map* $\mathcal{M}$ *such that* $\max(\mathcal{M}, \mathrm{OPT}) \leq \alpha$*, then* $w(\mathrm{GR}) \leq \alpha H_n \cdot w(\mathrm{OPT})$.

Proof. For every $\{i, j\} \in P(\mathrm{OPT})$, let $\mathcal{M}^{-1}(i, j) = \{e \in U : \mathcal{M}(e) = \{i, j\}\}$. By definition of $\mathcal{M}$, $\{\mathcal{M}^{-1}(i, j) : \{i, j\} \in P(\mathrm{OPT})\}$ is a partition of U. In each iteration of the algorithm, we distribute the cost of the newly picked object or pair of objects among the new covered elements: If x new elements are covered, each such element is charged $\frac{w_i}{x}$ or $\frac{w_i+w_j}{x}$, depending on whether a single object or a pair of objects are picked.

Let $\mathcal{M}^{-1}(i, j) = \{e'_1, \ldots, e'_k\}$, where the elements of $\mathcal{M}^{-1}(i, j)$ are indexed by the order they were first covered by the greedy algorithm, breaking ties arbitrarily. Consider the iteration in which e'_l was first covered. One possibility of the greedy algorithm was to pick $\{i, j\}$ (or if one of i and j was already picked, then take the other one), covering the elements $e'_l, \ldots, e'_k$, and possibly other elements as well. Therefore, each element that was covered in this iteration is charged at most $\frac{w_i+w_j}{k-l+1}$, and the total cost charged to the elements of $\mathcal{M}^{-1}(i, j)$ satisfies

$$\mathrm{charge}(\mathcal{M}^{-1}(i, j)) = \sum_{l=1}^{k} \mathrm{charge}(e'_l) \leq \sum_{l=1}^{k} \frac{w_i + w_j}{k - l + 1} \leq (w_i + w_j) H_n \ .$$

Since $w(\mathrm{GR})$ is charged to $e_1, \ldots, e_n$, we have

$$\begin{aligned} w(\mathrm{GR}) &= \sum_{j=1}^{n} \mathrm{charge}(e_j) \\ &= \sum_{\{i,j\} \in P(\mathrm{OPT})} \mathrm{charge}(\mathcal{M}^{-1}(i, j)) \\ &\leq H_n \sum_{\{i,j\} \in P(\mathrm{OPT})} (w_i + w_j) \mathbb{I}_{\mathcal{M}}(i, j) \\ &= H_n \sum_{i \in \mathrm{OPT}} w_i \sum_{j \neq i} \mathbb{I}_{\mathcal{M}}(i, j) \\ &\leq \alpha H_n \sum_{i \in \mathrm{OPT}} w_i \\ &= \alpha H_n \cdot w(\mathrm{OPT}) \ , \end{aligned}$$

where the last inequality holds since $\sum_{j \neq i} \mathbb{I}_{\mathcal{M}}(i, j) \leq \max(\mathcal{M}, \mathrm{OPT}) \leq \alpha$ for every $i \in \mathrm{OPT}$. □

Lemma 2. *If there exists an optimal solution* OPT *for cardinality SCP and a feasible map* $\mathcal{M}$ *such that* $\mathrm{mean}(\mathcal{M}, \mathrm{OPT}) \leq \alpha$*, then* $|\mathrm{GR}| \leq \alpha H_n \cdot |\mathrm{OPT}|$.

2.3 Approximation Ratio for Cardinality SCP

The conditional analysis in Lemma 2 is based on the existence of a feasible map $\mathcal{M}$ with respect to the optimal solution with small $\mathrm{mean}(\mathcal{M}, \mathrm{OPT})$. In Lemma 3 we demonstrate that there are instances of cardinality SCP in which a non-trivial map does not exist, and show that the approximation ratio of the greedy SCP algorithm might be $\Omega(\sqrt{n})$. However, in Theorem 4 we prove that the cardinality of the solution constructed by the greedy algorithm is within factor $\sqrt{2nH_n}$ of the minimum possible.

Lemma 3. *The approximation guarantee of the greedy algorithm for cardinality SCP is* $\Omega(\sqrt{n})$.

Theorem 4. $|\mathrm{GR}| \leq \sqrt{2nH_n} \cdot |\mathrm{OPT}|$.

Proof. We first observe that $|\mathrm{GR}| \leq 2n$, since in each iteration of the algorithm at least one element is covered using at most two objects. In addition, any feasible map $\mathcal{M}$ with respect to OPT certainly satisfies $\mathrm{mean}(\mathcal{M}, \mathrm{OPT}) \leq |\mathrm{OPT}|$, since

$$\mathrm{mean}(\mathcal{M}, \mathrm{OPT}) \leq \max(\mathcal{M}, \mathrm{OPT}) \leq |\mathrm{OPT}| \ .$$

By Lemma 2 we have $|\mathrm{GR}| \leq H_n \cdot |\mathrm{OPT}|^2$, and it follows that

$$|\mathrm{GR}| \leq \min\left\{2n, H_n \cdot |\mathrm{OPT}|^2\right\} \leq \left(2n\right)^{\frac{1}{2}} \left(H_n \cdot |\mathrm{OPT}|^2\right)^{\frac{1}{2}} = \sqrt{2nH_n} \cdot |\mathrm{OPT}| \ .$$

□

2.4 The Hardness of SCP: Additional Indications

The Case $|\mathcal{C}(i,j)| \leq 1$. The set cover problem becomes trivial when each subset contains a single element. However, in Theorem 5 we prove that SCP remains at least as hard to approximate as set cover when each pair of objects covers at most one element. We refer to this special case as SCP_1.

Theorem 5. *For any fixed $\epsilon > 0$, a polynomial time α-approximation algorithm for SCP_1 would imply a polynomial time $(1+\epsilon)\alpha$-approximation algorithm for set cover.*

Proof. Given a set cover instance I, with a ground set $U = \{e_1, \ldots, e_n\}$ and a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of U, we construct an instance $\rho(I)$ of SCP_1 as follows.

1. Let $k = \lceil \frac{n}{\epsilon} \rceil$.
2. The set of elements is $\bigcup_{t=1}^{k} \{e_1^t, \ldots, e_n^t\}$.
3. The set of objects is $(\bigcup_{t=1}^{k} \{S_1^t, \ldots, S_m^t\}) \cup \{y_1, \ldots, y_n\}$.
4. For $t = 1, \ldots, k$, $i = 1, \ldots, m$ and $j = 1, \ldots, n$, the pair $\{S_i^t, y_j\}$ covers e_j^t if $e_j \in S_i$.
5. Other pairs do not cover any element.

Let $\mathcal{S}^* \subseteq \mathcal{S}$ be a minimum cardinality set cover in I. Given a polynomial time α-approximation algorithm for SCP_1, we show how to find in polynomial time a set cover with cardinality at most $(1+\epsilon)\alpha|\mathcal{S}^*|$, for any fixed $\epsilon > 0$.

The construction of $\rho(I)$ guarantees that the collection of objects $\{S_i^t, y_1, \ldots, y_n\}$ covers the set of elements $\{e_j^t : e_j \in S_i\}$, for every $t = 1, \ldots, k$. Therefore, since $\mathcal{S}^*$ is a set cover in I, the objects $(\bigcup_{t=1}^k \{S_i^t : S_i \in \mathcal{S}^*\}) \cup \{y_1, \ldots, y_n\}$ cover all elements of $\rho(I)$. It follows that $\text{OPT}(\rho(I)) \leq k|\mathcal{S}^*| + n$, and we can find in polynomial time a feasible solution $\tilde{\mathcal{S}}$ to $\rho(I)$ such that $|\tilde{\mathcal{S}}| \leq \alpha(k|\mathcal{S}^*|+n)$. Let t' be the index t for which $|\tilde{\mathcal{S}} \cap \{S_1^t, \ldots, S_m^t\}|$ is minimized. Then $\mathcal{S}' = \{S_i : S_i^{t'} \in \tilde{\mathcal{S}} \cap \{S_1^{t'}, \ldots, S_m^{t'}\}\}$ is a set cover in I with cardinality

$$|\mathcal{S}'| = \min_{t=1,\ldots,k} |\tilde{\mathcal{S}} \cap \{S_1^t, \ldots, S_m^t\}| \leq \frac{|\tilde{\mathcal{S}}|}{k} \leq \frac{\alpha(k|\mathcal{S}^*|+n)}{k} \leq (1+\epsilon)\alpha|\mathcal{S}^*| \ .$$

□

Integrality Gap of LP-Relaxation. In contrast with the set cover problem, for which the integrality gap of the natural LP-relaxation is $O(\log n)$ [13], we show in Theorem 6 that the integrality gap of the corresponding relaxation of SCP is $\Omega(n)$.

SCP can be formulated as an integer program by:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in \mathcal{S}} w_i x_i & & \\ \text{subject to} \quad & \sum_{\{i,j\}: e \in \mathcal{C}(i,j)} y_{\{i,j\}} \geq 1 & \forall e \in U & \quad (2.1) \\ & y_{\{i,j\}} \leq x_i & \forall i \neq j \in \mathcal{S} & \quad (2.2) \\ & x_i,\, y_{\{i,j\}} \in \{0,1\} & \forall i \neq j \in \mathcal{S} & \quad (2.3) \end{aligned}$$

The variable x_i indicates whether the object i is chosen for the cover, whereas $y_{\{i,j\}}$ indicates whether both i and j are chosen. Constraint (2.1) guarantees that for each element $e \in U$ we pick at least one pair of objects that covers it. Constraint (2.2) ensures that a pair of objects cannot cover any element unless we indeed pick both objects. The LP-relaxation of this integer program, (LP), is obtained by replacing the integrality constraint (2.3) with $x_i \geq 0$ and $y_{\{i,j\}} \geq 0$.

Theorem 6. *The integrality gap of* (LP) *is* $\Omega(n)$*, even for cardinality SCP.*

Proof. Consider the instance of cardinality SCP with $U = \{e_1, \ldots, e_n\}$ and $\mathcal{S} = \{1, \ldots, 2n\}$. The elements covered by pairs of objects in $\mathcal{S}$ are:

1. $\mathcal{C}(i, n+1) = \mathcal{C}(i, n+2) = \cdots = \mathcal{C}(i, 2n) = \{e_i\}$, $i = 1, \ldots, n$.
2. Other pairs do not cover any element.

Since any integral solution must pick the objects $1, \ldots, n$ and at least one of the objects $n+1, \ldots, 2n$, $|\text{OPT}| \geq n+1$. We claim that the fractional solution $x_i' = \frac{1}{n}$ and $y_{\{i,j\}}' = \frac{1}{n}$ for every $i \neq j$ is feasible for (LP). Clearly, this solution is non-negative and satisfies constraint (2.2). In addition, $\sum_{\{i,j\}:e\in\mathcal{C}(i,j)} y'_{\{i,j\}} = 1$ for every $e \in U$, since the left-hand-side contains exactly n summands, each of value $\frac{1}{n}$. It follows that the cost of an optimal fractional solution is at most 2, and the integrality gap of (LP) is at least $\frac{n+1}{2}$. □

3 Remote Dominating Set

In the following we show that there are instances of the problem in which a non-trivial map does not exist, and demonstrate that the greedy algorithm might construct a solution for RDS whose cost is $\Omega(\sqrt{n})$ times the optimum. On the positive side however, we provide a $2H_n$-approximation algorithm for RDS that constructs a remote dominating set by approximating two dependent set cover problems.

3.1 The Greedy SCP Algorithm for RDS

According to our interpretation of the RDS problem as a special case of SCP, the greedy algorithm picks in each iteration a single point or a pair of points, whichever attains the minimum ratio of cost to number of newly covered points. By modifying the construction in Lemma 3, we prove in Lemma 7 that the approximation ratio of this algorithm is $\Omega(\sqrt{n})$.

Lemma 7. *The approximation guarantee of the greedy algorithm for RDS is* $\Omega(\sqrt{n})$.

3.2 A $2H_n$-Approximation Algorithm

Despite these negative results regarding the performance of the greedy SCP algorithm for the RDS problem, we show that this problem can still be approximated to within a logarithmic factor. Our algorithm constructs a remote dominating set by approximating two dependent set cover problems, (SC_1) and (SC_2).

For $v \in V$, let $N_v = \{u \in V : d(v,u) \leq r_2\}$. Using the greedy set cover algorithm, we construct an RDS in two phases:

1. We first approximate (SC_1): The set of elements to cover is V; the covering sets are $\mathcal{S} = \{N_v : v \in V\}$; the cost of N_v is c_v. Let S_1 be the cover we obtain. V can now be partitioned into two sets: V_1, points within distance r_1 of some point in S_1 or within distance r_2 of two points in S_1, and $V_2 = V \setminus V_1$.
2. We then approximate (SC_2): The set of elements to cover is V_2; the covering sets are $\mathcal{S} = \{N_v : v \in V \setminus S_1\}$; the cost of N_v is c_v. Let S_2 be the cover we obtain.

Theorem 8. *Let* OPT *be a minimum cost RDS. Then*

1. $S_1 \cup S_2$ *is an RDS.*
2. $c(S_1 \cup S_2) \leq 2H_n \cdot c(\mathrm{OPT})$.

4 Cardinality RDS on a Tree

In this section we consider the minimum cardinality RDS problem on a tree $T = (V, E)$ with unit length edges and covering radii $r_1 = 1$ and $r_2 = 2$. It would be convenient to work directly with the tree representation of the problem, instead of working with the related metric space.

We constructively show that a minimum cardinality dominating set in T is a 2-approximation, by exploiting special properties of a partial map we find. We also prove that this bound is tight, and demonstrate that in general graphs a minimum cardinality dominating set does not guarantee a non-trivial approximation ratio.

4.1 The Existence of Acyclic Mapping Graphs

Let S be an RDS that contains at least two vertices. We denote by $L \subseteq V$ the set of vertices that are not covered by a single vertex in S. In other words, $v \in L$ if there is no vertex in S within distance 1 of v. Given a partial map $\mathcal{M}_L : L \to P(S)$, its *mapping graph* $\mathcal{G}(\mathcal{M}_L)$ is defined by:

1. The set of vertices of $\mathcal{G}(\mathcal{M}_L)$ is S.
2. For $u \neq v \in S$, (u, v) is an edge of $\mathcal{G}(\mathcal{M}_L)$ if there is a vertex $w \in L$ such that $\mathcal{M}_L(w) = \{u, v\}$.

Lemma 9. *There is a partial map $\mathcal{M}_L : L \to P(S)$ whose mapping graph $\mathcal{G}(\mathcal{M}_L)$ is acyclic.*

4.2 A 2-Approximation Algorithm

Based on the existence of a partial map whose mapping graph is acyclic, in Lemma 10 we constructively show that for every remote dominating set S in T there is a dominating set of cardinality at most $2|S| - 1$.

Lemma 10. *Let S be an RDS in T. Then there is a dominating set of cardinality at most* $2|S| - 1$.

A minimum cardinality dominating set D^* in T can be found in linear time [2], and in the special case we consider, this set is also an RDS. Lemma 10 proves, in particular, the existence of a dominating set whose cardinality is at most $2|\mathrm{OPT}| - 1$, where OPT is a minimum cardinality RDS in T. We have as a conclusion the following theorem.

Theorem 11. $|D^*| \leq 2|\mathrm{OPT}| - 1$.

In Lemma 12 we show that the bound given in Theorem 11 is tight, by providing an instance with $|D^*| = 2|\mathrm{OPT}| - 1$. We also demonstrate that in general graphs a minimum cardinality dominating set does not guarantee a non-trivial approximation ratio.

Lemma 12. *There are instances in which $|D^*| = 2|\mathrm{OPT}| - 1$. In addition, when the underlying graph is not restricted to be a tree, there are instances with $|\mathrm{OPT}| = O(1)$ and $|D^*| = \Omega(n)$.*

5 Euclidean RDS

In this section we present a polynomial time approximation scheme for the Euclidean RDS problem, following the general framework suggested by Hochbaum and Maass for covering and packing problems in Euclidean spaces [10]. The unifying idea behind their *shifting* strategy is to repeatedly apply a simple divide-and-conquer approach and select the best solution we find.

To simplify the presentation, we denote by $P = \{p_1, \ldots, p_n\}$ the set of points to be covered, and let $D = r_2$. We also assume that P is bounded in a rectangle I, where the length of the long edge of I is nD. Otherwise, we can partition P into sets for which this property is satisfied, and separately use the algorithm for each set.

The Vertical Partitions. We divide I into pairwise disjoint vertical strips of width D. Given a shifting parameter l, the partition V_0 of I consists of strips of width lD. For every $i = 1, \ldots, l-1$, let V_i be the partition of I obtained by shifting V_0 to the right over distance iD.

For each partition V_i we define a set of points $\mathrm{OPT}(V_i)$ as follows. For every strip J in the partition V_i, let $\mathrm{OPT}(V_i, J)$ be a minimum cardinality set of points in P that covers the points P_J, where P_J is the set of points in P located in the strip J. Then $\mathrm{OPT}(V_i) = \bigcup_{J \in V_i} \mathrm{OPT}(V_i, J)$. Clearly, $\mathrm{OPT}(V_i)$ is an RDS.

Lemma 13. *Let* OPT *be a minimum cardinality Euclidean RDS, then*

$$\min_{i=0,\ldots,l-1} |\mathrm{OPT}(V_i)| \leq \left(1 + \frac{2}{l}\right) |\mathrm{OPT}| \ .$$

The Horizontal Partitions. We are now concerned with the problem of finding a small set of points in P that covers P_J, for a given strip J. We divide J into pairwise disjoint horizontal strips of height D. The partition H_0 of J consists of strips of height lD. For every $i = 1, \ldots, l-1$, let H_i be the partition of J obtained by shifting H_0 up over distance iD.

For each partition H_i we define a set of points $\mathrm{OPT}(H_i)$ as follows. For every strip R in the partition H_i, let $\mathrm{OPT}(H_i, R)$ be a minimum cardinality set of points in P that covers the points P_R, where P_R is the set of points in P located in the strip R. Then $\mathrm{OPT}(H_i) = \bigcup_{R \in H_i} \mathrm{OPT}(H_i, R)$. Clearly, $\mathrm{OPT}(H_i)$ is a set of points in P that covers P_J.

Lemma 14. *Let* OPT^J *be a minimum cardinality set of points in* P *that covers* P_J*, then*

$$\min_{i=0,\ldots,l-1} |\mathrm{OPT}(H_i)| \leq \left(1 + \frac{2}{l}\right) |\mathrm{OPT}^J| \ .$$

Optimal Solution in an $lD \times lD$ Square. Lemmas 13 and 14 show that in order to obtain a polynomial time approximation scheme for Euclidean RDS, it is sufficient to optimally solve the following problem: Given R, an $lD \times lD$ square in I, find a minimum cardinality set of points in P that covers P_R. The next lemma allows us to perform an exhaustive search for an optimal solution to this problem in time $O(n^{O(l^2)})$.

Lemma 15. *There is a set of points $S \subseteq P$, $|S| = O(l^2)$, that covers P_R.*

Theorem 16. *There is a polynomial time approximation scheme for the Euclidean RDS problem.*

6 Group Cut on a Path

In this section we first discuss the hardness of approximating GCP, and prove that this problem is as hard to approximate as set cover. We also identify the exact point at which GCP becomes NP-hard. We then present a simple proof for the existence of a feasible map $\mathcal{M}$ with respect to the optimal solution for which $\max(\mathcal{M}, \mathrm{OPT}) \leq 2$. This result, combined with Lemma 1, enables us to show that the approximation ratio of the greedy SCP algorithm for this special case is $2H_k$.

6.1 Hardness Results

By describing an approximation preserving reduction, we prove in Theorem 17 that GCP is as hard to approximate as set cover. A special case of this reduction also shows that GCP is as hard to approximate as vertex cover even when the cardinality of each group is at most 4. In addition, we prove in Lemma 18 that when the bound on cardinality is 3, the problem is polynomial time solvable.

Theorem 17. *A polynomial time approximation algorithm for the GCP problem with factor α would imply a polynomial time approximation algorithm for the set cover problem with the same factor.*

Note that vertex cover is a special case of set cover in which each element belongs to exactly two sets. Therefore, the proof of Theorem 17 can be modified to show that GCP is as hard to approximate as vertex cover even when the cardinality of each group is at most 4.

Lemma 18. *GCP is polynomial time solvable when $|G_i| \leq 3$ for every $i = 1, \ldots, k$.*

6.2 A Feasible Map with Small Max

Let $F \subseteq E$ be any feasible solution, with $|F| \geq 2$. In Lemma 19 we prove the existence of a feasible map $\mathcal{M} : \{G_1, \ldots, G_k\} \to P(F)$ for which $\max(\mathcal{M}, F) \leq 2$.

Lemma 19. *There is a feasible map $\mathcal{M} : \{G_1, \ldots, G_k\} \to P(F)$ with*

$$\max(\mathcal{M}, F) \leq 2 \ .$$

Let OPT be a minimum cost set of edges that separates $G_1, \ldots, G_k$, and without loss of generality $|\mathrm{OPT}| \geq 2$. The next theorem follows from Lemmas 1 and 19.

Theorem 20. *The greedy SCP algorithm constructs a solution whose cost is at most $2H_k \cdot c(\mathrm{OPT})$.*

7 Concluding Remarks

There is a huge gap between the upper bound for approximating the cardinality SCP problem, that was established in Theorem 4, and the logarithmic lower bound that follows from the observation that SCP contains set cover as a special case. The first, and probably the most challenging, open problem is to obtain either an improved hardness result or an improved approximation algorithm. Another open problem in this context is to provide a non-trivial algorithm for the general problem.

In addition, it would be interesting to study the seemingly simple special case, in which each pair of objects covers at most one element. We proved that this problem is at least as hard to approximate as set cover, but we do not know how to significantly improve the approximation guarantee. Moreover, we consider this case to demonstrate the main difficulty in approximating SCP, as it shows that the objective is to choose a dense set of objects that covers all elements.

We suggest for future research the *partial SCP* problem, a variant of SCP in which we are given an additional parameter k, and the objective is to cover at least k elements with minimum cost. This problem is closely related to the *dense k-subgraph* problem, that required to find in a given graph $G = (V, E)$ a set of k vertices whose induced subgraph has maximum number of edges. This problem is NP-hard, and the currently best approximation guarantee in general graphs is $O(n^{-\delta})$, for some constant $\delta < \frac{1}{3}$, due to Feige, Kortsarz and Peleg [5]. The next theorem relates these problems, and shows that the approximation guarantee of dense k-subgraph can be improved by developing an $o(n^{\delta/2})$-approximation algorithm for partial SCP.

Theorem 21. *A polynomial time $\alpha(k)$-approximation algorithm for partial SCP would imply a randomized polynomial time $\frac{1}{\alpha^2(k^2)(1+\epsilon)}$-approximation algorithm for dense k-subgraph, for any fixed $\epsilon > 0$.*

References

1. V. Chvátal. A greedy heuristic for the set covering problem. *Mathematics of Operations Research*, 4:233–235, 1979.
2. E. J. Cockayne, S. E. Goodman, and S. T. Hedetniemi. A linear algorithm for the domination number of a tree. *Information Processing Letters*, 4:41–44, 1975.
3. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.
4. U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.
5. U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29:410–421, 2001.
6. N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18:3–20, 1997.
7. S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, 1998.
8. R. Hassin and D. Segev. The set cover with pairs problem, 2005. http://www.math.tau.ac.il/∼segevd/Papers/SCP-Jour.pdf.

9. R. Hassin and A. Tamir. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10:395–402, 1991.
10. D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32:130–136, 1985.
11. D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.
12. F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.
13. L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.
14. P. Slavík. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64:251–254, 1997.