A Unified Approach to Approximating Partial Covering Problems*

Jochen Könemann† Ojas Parekh‡ Danny Segev§

Abstract

An instance of the *generalized partial cover* problem consists of a ground set U and a family of subsets $\mathcal{S} \subseteq 2^U$. Each element $e \in U$ is associated with a profit $p(e)$, whereas each subset $S \in \mathcal{S}$ has a cost $c(S)$. The objective is to find a minimum cost subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that the combined profit of the elements covered by $\mathcal{S}'$ is at least P, a specified profit bound. In the *prize-collecting* version of this problem, there is no strict requirement to cover any element; however, if the subsets we pick leave an element $e \in U$ uncovered, we incur a penalty of $\pi(e)$. The goal is to identify a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ that minimizes the cost of $\mathcal{S}'$ plus the penalties of uncovered elements.

Although problem-specific connections between the partial cover and the prize-collecting variants of a given covering problem have been explored and exploited, a more general connection remained open. The main contribution of this paper is to establish a formal relationship between these two variants. As a result, we present a unified framework for approximating problems that can be formulated or interpreted as special cases of generalized partial cover. We demonstrate the applicability of our method on a diverse collection of covering problems, for some of which we obtain the first non-trivial approximability results.

Keywords: Partial cover, approximation algorithms, Lagrangian relaxation.

1 Introduction

For over three decades the *set cover* problem and its ever-growing list of generalizations, variants, and special cases have attracted the attention of researchers in the fields of discrete optimization, complexity theory, and combinatorics. Essentially, these problems are concerned with identifying a minimum cost collection of sets that covers a given set of elements, possibly with additional side constraints. While such settings may appear to be very simple at first glance, they still capture computational tasks of great theoretical and practical importance, as the reader may verify by consulting directly related surveys [2, 15, 22, 33] and the references therein.

In the present paper we focus our attention on the *generalized partial cover* problem, whose input consists of a ground set of elements U and a family $\mathcal{S}$ of subsets of U. In addition, each element $e \in U$ is associated with a profit $p(e)$, whereas each subset $S \in \mathcal{S}$ has a cost $c(S)$. The objective is to find a minimum cost subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that the combined profit of the elements covered by

*An extended abstract of this paper appeared in *Proceedings of the 14th Annual European Symposium on Algorithms*, 2006.

†Department of Combinatorics and Optimization, University of Waterloo, Waterloo ON N2L 3G1, Canada. Email: jochen@math.uwaterloo.ca. Research supported by NSERC grant no. 288340-2004.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA. Email: ojas@mathcs.emory.edu.

§School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: segevd@post.tau.ac.il.

$\mathcal{S}'$ is at least P, a specified profit bound. When all elements are endowed with unit profits, we obtain the well-known *partial cover* problem, in which the goal is to cover a given number of elements by picking subsets of minimum total cost.

Numerous computational problems can be formulated or interpreted as special cases of generalized partial cover, although this fact may be well-hidden. For most of these problems, novel techniques in the design of approximation algorithms have emerged over the years, and it is clearly beyond the scope of this writing to present an exhaustive overview. However, from the abundance of greedy schemes, local-search heuristics, randomized methods, and LP-based algorithms a simple observation is revealed: There is currently no unified approach to approximating partial covering problems.

1.1 The suggested method

Preliminaries. The main contribution of this paper is to establish a formal relationship between the partial cover and the prize-collecting versions of a given covering problem. In the *prize-collecting set cover* problem there is no strict requirement to cover any element; however, if the subsets we pick leave an element $e \in U$ uncovered, we incur a penalty of $\pi(e)$. The objective is to find a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ that minimizes the cost of $\mathcal{S}'$ plus the penalties of the uncovered elements. A polynomial-time algorithm for this problem is said to be *Lagrangian multiplier preserving with factor r* (henceforth, r-LMP) if for every instance I it constructs a solution that satisfies $C + r\Pi \leq r \cdot \mathrm{OPT}(I)$, where C is the total cost of the subsets picked, and Π is the sum of penalties over all uncovered elements. We further denote by $\mathcal{I}_r$ the family of weighted set systems $(U, \mathcal{S}, c)$ that possess the following property: There is an r-LMP algorithm for all prize-collecting instances $(U, \mathcal{S}, c, \pi)$, $\pi : U \to \mathbb{Q}_+$. In other words, for every penalty function π the corresponding instance admits an r-LMP approximation.

The main result. At the heart of our method is an algorithm for the generalized partial cover problem that computes an approximate solution by making use of an r-LMP prize-collecting algorithm in a black-box fashion. Specifically, in Section 2 we prove the following theorem.

Theorem 1. *Let I be a generalized partial cover instance defined on an underlying weighted set system $(U, \mathcal{S}, c)$, and suppose that $(U, \mathcal{S}, c) \in \mathcal{I}_r$ for some $r \geq 1$. Then, for any $\epsilon > 0$, we can find a feasible solution to I whose cost is at most $(\frac{4}{3} + \epsilon)r$ times the optimum, within time polynomial in $|U|$, $|\mathcal{S}|^{1/\epsilon}$ and the input length of I.*

To simplify the presentation, it is convenient to state our approach in terms of the Lagrangian relaxation technique, which was first utilized in the context of approximation algorithms by Jain and Vazirani [24]. Here is a rough outline of how the proof of Theorem 1 will proceed. We begin by formulating the generalized partial cover problem as an integer program. Next, we dualize the complicating constraint that places a lower bound of P on the total profit. More precisely, we lift this constraint to the objective function multiplied by an auxiliary variable λ, and obtain its corresponding Lagrangian relaxation. For any fixed $\lambda \geq 0$, the new program describes, up to a constant term, a prize-collecting set cover instance with non-uniform penalties. We now conduct a binary search, using the r-LMP prize-collecting algorithm as a subroutine, to find sufficiently close $\lambda_1 \geq \lambda_2$ that satisfy: For λ_1, the algorithm constructs a solution $\mathcal{S}_1 \subseteq \mathcal{S}$ such that the total profit of the elements covered by $\mathcal{S}_1$ is at least P; For λ_2, it constructs a solution $\mathcal{S}_2 \subseteq \mathcal{S}$ with a total profit of at most P.

Although we can exploit the r-LMP property to show that the cost of $\mathcal{S}_2$ is within factor r of optimum, this solution is not necessarily feasible. The situation is quite the opposite with respect to $\mathcal{S}_1$, which is a feasible solution whose cost may be arbitrarily large. Having observed these facts, we create an additional feasible solution $\mathcal{S}_3$ by augmenting $\mathcal{S}_2$ with a carefully chosen subset of $\mathcal{S}_1$. The cost of this subset is bounded by extending the arguments used by Levin and Segev [28] and independently by Golovin, Nagarajan and Singh [20] for approximating the *k-multicut* problem. Finally, we establish Theorem 1 by proving that the cost of the cheaper of $\mathcal{S}_1$ and $\mathcal{S}_3$ is at most $(\frac{4}{3}+\epsilon)r$ times the cost of an optimal solution.

1.2 Designing LMP algorithms

At this point in time, the reader should bear in mind that the performance guarantee of our algorithm, as stated in Theorem 1, depends on the existence of an LMP prize-collecting algorithm for a given covering problem. Indeed, this dependence appears to be the primary factor limiting the employment of Lagrangian relaxations in most problems of interest. The latter drawback was pointed out by Chudak, Roughgarden and Williamson [8], who asked whether it is possible to devise more general variants of the Lagrangian relaxation framework that apply to a broader class of problems. We answer this question in the affirmative, by developing prize-collecting algorithms with the LMP property for some of the most fundamental integer covering problems. These results, along with a detailed description of previous work, are formally presented in Sections 3 and 4.

Even though the algorithms we suggest are rather problem-specific, a principal idea is brought into play in the majority of the applications we consider. Intuitively, ensuring an inequality of the form $C + r\Pi \leq r \cdot \mathrm{OPT}$ means that the solutions we construct are efficient when it comes to paying penalties. Technically speaking, such a solution guarantees an r-approximation even when all penalties are inflated by a factor of r. Given a prize-collecting instance, our general approach is to create a new instance of the underlying full coverage problem, in which penalties are represented as alternative covering options with inflated costs. We then propose a tailor-made algorithm, or modify the analysis of an existing one, to find a solution to the resulting instance, and show that it can be interpreted as an approximate solution to the original problem.

2 The Generalized Partial Cover Algorithm

The main result of this section is a constructive proof of Theorem 1. Recall that a generalized partial cover instance I is defined with respect to an underlying weighted set system, consisting of a ground set U and a family of subsets $\mathcal{S} \subseteq 2^U$, where each $S \in \mathcal{S}$ has a cost $c(S)$. The additional ingredients of I are profits $p(e)$, specified for each element $e \in U$, and a requirement parameter P. Now suppose that $(U, \mathcal{S}, c) \in \mathcal{I}_r$ for some $r \geq 1$, meaning that there is an r-LMP algorithm $\mathcal{A}$ for all prize-collecting instances $(U, \mathcal{S}, c, \pi)$, $\pi : U \to \mathbb{Q}_+$.

2.1 Preliminaries

The method we suggest and its analysis will be based on a natural integer programming formulation of the generalized partial cover problem. In the following, let $\mathcal{S}_e \subseteq \mathcal{S}$ be the collection of sets that contain $e \in U$, and let $P_U = \sum_{e \in U} p(e)$.

$$\begin{aligned}
\text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S && && \text{(GC)} \\
\text{subject to} \quad & \sum_{S \in \mathcal{S}_e} x_S + z_e \geq 1 && \forall\, e \in U && (2.1) \\
& \sum_{e \in U} p(e) z_e \leq P_U - P && && (2.2) \\
& x_S,\, z_e \in \{0, 1\} && \forall\, S \in \mathcal{S},\, e \in U && (2.3)
\end{aligned}$$

In this formulation, the variable x_S indicates whether we pick the set S, whereas z_e indicates whether the element e is uncovered. Constraint (2.1) guarantees that we either pick at least one set that contains e, or specify that this element is uncovered by setting $z_e = 1$. Constraint (2.2) forces any feasible solution to cover elements with a total profit of at least P.

Essential to the subsequent analysis will be the fact that the LP-relaxation of (GC), obtained by replacing constraint (2.3) with $x_S \geq 0$ and $z_e \geq 0$, has an integrality gap of $O(r)$. Unfortunately, this prerequisite is not satisfied even in the case of unit profits, as the next example illustrates. Consider an instance in which the ground set U consists of n elements, and the family $\mathcal{S}$ contains a single set $S = U$ with cost n. When we are required to cover at least one element, the integral optimum is clearly n. However, by setting $x_S = \frac{1}{n}$ and $z_e = 1 - \frac{1}{n}$ for every $e \in U$, we define a feasible fractional solution whose cost is 1. This example, as well as additional constructions of similar nature, demonstrate that an unbounded integrality gap may arise whenever a small number of sets in the optimal solution contribute a large fraction of its cost.

Therefore, an inevitable part of our algorithm is a preprocessing step in which, given a fixed accuracy parameter $\epsilon > 0$, we "guess" the $\lfloor \frac{1}{\epsilon} \rfloor$ most expensive sets in the optimal solution, whose cost we denote by OPT. More precisely, we enumerate all $O(|\mathcal{S}|^{1/\epsilon})$ subsets $\mathcal{S}' \subseteq \mathcal{S}$ of cardinality at most $\lfloor \frac{1}{\epsilon} \rfloor$, test each such subset as the correct guess, and return the best solution we find. For a given subset $\mathcal{S}'$, we include it as part of the solution to be constructed, eliminate the sets in $\mathcal{S}'$ from $\mathcal{S}$, remove all covered elements from U and from the remaining sets, and update the profit requirement. Any set whose cost is greater than $\min_{S \in \mathcal{S}'} c(S)$ is also eliminated. Consequently, the cost of each remaining set is at most $\epsilon \cdot \text{OPT}$.

In the remainder of this section we will bypass the preprocessing step, and assume that the maximum cost of a set in $\mathcal{S}$ is at most $\epsilon \cdot \text{OPT}$. For ease of presentation, we also assume that $c(S) > 0$ for every $S \in \mathcal{S}$ and that $p(e) > 0$ for every $e \in U$, since zero-cost sets can be picked in advance and zero-profit elements can be discarded.

2.2 Obtaining $\mathcal{S}_1$ and $\mathcal{S}_2$

We now dualize the profit constraint (2.2), and lift it to the objective function multiplied by $\lambda \geq 0$. The resulting Lagrangian relaxation is:

$$\begin{aligned}
\text{LR}(\lambda) = \text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S + \lambda \left(\sum_{e \in U} p(e) z_e - (P_U - P) \right) \\
\text{subject to} \quad & \sum_{S \in \mathcal{S}_e} x_S + z_e \geq 1 \qquad \forall\, e \in U \\
& x_S,\, z_e \in \{0, 1\} \qquad \forall\, S \in \mathcal{S},\, e \in U
\end{aligned}$$

We remark that, excluding the constant term of $-\lambda(P_U - P)$ in the objective function, $\mathrm{LR}(\lambda)$ is an integer programming formulation of the prize-collecting set cover problem, in which each element $e \in U$ is associated with a penalty $\lambda p(e)$. We refer to this instance as I_λ, and use $\mathrm{OPT}(I_\lambda)$ to denote its optimum value. It is not difficult to verify that $\mathrm{LR}(\lambda) = \mathrm{OPT}(I_\lambda) - \lambda(P_U - P)$ is at most OPT for any $\lambda \geq 0$, by observing that an optimal solution to (GC) is also a feasible solution to $\mathrm{LR}(\lambda)$, whose cost is at most OPT.

Since the underlying weighted set system of I_λ is identical to that of I, we may apply the prize-collecting algorithm $\mathcal{A}$ to approximate I_λ. Let x^λ indicate which sets in $\mathcal{S}$ were picked by the algorithm, and let z^λ indicate which elements were left uncovered. In terms of (x^λ, z^λ), the r-LMP property of $\mathcal{A}$ is equivalent to

$$\sum_{S \in \mathcal{S}} c(S) x_S^\lambda + r \sum_{e \in U} \lambda p(e) z_e^\lambda \leq r \cdot \mathrm{OPT}(I_\lambda) \ , \tag{2.4}$$

an inequality that, in particular, leads to the following observation.

Lemma 2. *When $\lambda > \frac{1}{\min_{e \in U} p(e)} \sum_{S \in \mathcal{S}} c(S)$, the solution (x^λ, z^λ) covers all elements. On the other hand, (x^0, z^0) does not cover any element.*

Proof. Let $\lambda > \frac{1}{\min_{e \in U} p(e)} \sum_{S \in \mathcal{S}} c(S)$, and suppose that there is an element $\bar{e} \in U$ for which $z_{\bar{e}}^\lambda = 1$, that is, $\bar{e}$ is not covered by any set the algorithm $\mathcal{A}$ picks when we approximate I_λ. Then (x^λ, z^λ) no longer satisfies inequality (2.4), as

$$\sum_{S \in \mathcal{S}} c(S) x_S^\lambda + r \sum_{e \in U} \lambda p(e) z_e^\lambda \geq r \lambda p(\bar{e}) > r \frac{p(\bar{e})}{\min_{e \in U} p(e)} \sum_{S \in \mathcal{S}} c(S) \geq r \sum_{S \in \mathcal{S}} c(S) \geq r \cdot \mathrm{OPT}(I_\lambda) \ ,$$

where the last inequality holds since $\mathcal{S}$ is a feasible solution to I_λ.

Now let $\lambda = 0$, and note that each element of the instance I_0 has a zero penalty. Therefore, by deciding not to pick any set and instead pay all penalties we obtain a feasible solution with zero cost, implying that $\mathrm{OPT}(I_0) = 0$. Since (x^0, z^0) satisfies inequality (2.4), it follows that this solution cannot pick any set, as all sets in $\mathcal{S}$ have strictly positive costs by assumption. ∎

This observation allows us to conduct a binary search over the interval $[0, \frac{2}{\min_e p(e)} \sum_{S \in \mathcal{S}} c(S)]$, consisting of a polynomially-bounded number of calls to the prize-collecting algorithm $\mathcal{A}$, as a result of which we find $\lambda_1 \geq \lambda_2$ that satisfy:

1. $\lambda_1 - \lambda_2 \leq \frac{\epsilon c_{\min}}{P_U}$, where $c_{\min} = \min_{S \in \mathcal{S}} c(S) > 0$.
2. The elements covered by $(x^{\lambda_1}, z^{\lambda_1})$ have a total profit of $P_1 \geq P$, and at the same time those covered by $(x^{\lambda_2}, z^{\lambda_2})$ have a total profit of $P_2 \leq P$.

For ease of notation, we designate by $\mathcal{S}_1$ and $\mathcal{S}_2$ the subsets of $\mathcal{S}$ that were picked by the solutions $(x^{\lambda_1}, z^{\lambda_1})$ and $(x^{\lambda_2}, z^{\lambda_2})$, respectively. Without loss of generality, $P_1 > P$, or otherwise $\mathcal{S}_1$ is already a feasible solution whose cost is at most $r \cdot \mathrm{LR}(\lambda_1) \leq r \cdot \mathrm{OPT}$. Similarly, we assume that $P_2 < P$. The analysis of our algorithm crucially depends on the next lemma, which is a consequence of the r-LMP property.

Lemma 3. *Let $\alpha = \frac{P - P_2}{P_1 - P_2} \in (0, 1)$. Then, $\alpha c(\mathcal{S}_1) + (1 - \alpha) c(\mathcal{S}_2) \leq r(1 + \epsilon)\mathrm{OPT}$.*

Proof. By combining inequality (2.4) with the fact that $\mathrm{LR}(\lambda) = \mathrm{OPT}(I_\lambda) - \lambda(P_U - P) \le \mathrm{OPT}$ for every $\lambda \ge 0$, we have

$$\begin{aligned} c(\mathcal{S}_1) &= \sum_{S \in \mathcal{S}} c(S) x_S^{\lambda_1} \\ &\le r \left(\mathrm{OPT}(I_{\lambda_1}) - \lambda_1 \sum_{e \in U} p(e) z_e^{\lambda_1} \right) \\ &= r \left(\mathrm{OPT}(I_{\lambda_1}) - \lambda_1 (P_U - P_1) \right) \\ &= r(\mathrm{LR}(\lambda_1) + \lambda_1(P_1 - P)) \\ &\le r(\mathrm{OPT} + \lambda_1(P_1 - P)) \ . \end{aligned} \tag{2.5}$$

A similar argument shows that $c(\mathcal{S}_2) \le r(\mathrm{OPT} + \lambda_2(P_2 - P))$. Therefore,

$$\begin{aligned} \alpha c(\mathcal{S}_1) + (1-\alpha) c(\mathcal{S}_2) &\le \alpha r(\mathrm{OPT} + \lambda_1(P_1 - P)) + (1-\alpha) r(\mathrm{OPT} + \lambda_2(P_2 - P)) \\ &\le r \cdot \mathrm{OPT} + \alpha r \left(\lambda_2 + \frac{\epsilon c_{\min}}{P_U} \right) (P_1 - P) + (1-\alpha) r \lambda_2 (P_2 - P) \\ &= r \cdot \mathrm{OPT} + r\lambda_2 \left(\alpha(P_1 - P) + (1-\alpha)(P_2 - P) \right) + r\alpha\epsilon c_{\min} \cdot \frac{P_1 - P}{P_U} \\ &\le r \cdot \mathrm{OPT} + r\epsilon c_{\min} \\ &\le r(1+\epsilon)\mathrm{OPT} \ . \end{aligned}$$

The second inequality follows from observing that $P_1 > P$ and $\lambda_1 \le \lambda_2 + \frac{\epsilon c_{\min}}{P_U}$. The third inequality holds since $\alpha(P_1 - P) + (1-\alpha)(P_2 - P) = 0$, $\alpha < 1$ and $P_1 - P \le P_U$. ■

2.3 Composing an additional solution

Up until now, the only feasible solution we have at our possession is $\mathcal{S}_1$, as this subset of $\mathcal{S}$ covers elements with an overall profit of $P_1 > P$. Inequality (2.5) places an upper bound of $r \cdot \mathrm{OPT} + r\lambda_1(P_1 - P)$ on the cost of $\mathcal{S}_1$. However, the latter term may be arbitrarily large in comparison to OPT, implying that $\mathcal{S}_1$ cannot approximate the instance I by itself. The situation is quite the opposite with respect to $\mathcal{S}_2$: Although this solution covers elements with an insufficient profit of $P_2 < P$, a similar bound of $r \cdot \mathrm{OPT} + r\lambda_2(P_2 - P)$ on its cost actually yields the inequality $c(\mathcal{S}_2) \le r \cdot \mathrm{OPT}$, since in this case $r\lambda_2(P_2 - P) \le 0$.

At this point, we are concerned with creating an additional feasible solution $\mathcal{S}_3$, by augmenting $\mathcal{S}_2$ with a carefully chosen subset $\mathcal{S}' \subseteq \mathcal{S}_1$. To attain feasibility, we must ensure that of the elements that were left uncovered by $\mathcal{S}_2$, a subcollection with a total profit of at least $P - P_2$ is covered by $\mathcal{S}'$. We construct this augmenting subset as follows. Let $U' \subseteq U$ be the collection of elements that are covered by $\mathcal{S}_1$ but not by $\mathcal{S}_2$. We assign each element $e \in U'$ to an arbitrary set in $\mathcal{S}_1 \setminus \mathcal{S}_2$ that contains it, and denote by $\varphi(S)$ the total profit of the elements assigned to S. Without loss of generality, we assume that $\mathcal{S}_1 \setminus \mathcal{S}_2 = \{S_1, \ldots, S_k\}$, where these sets are indexed by non-decreasing order of the ratio $\frac{c(S_i)}{\varphi(S_i)}$. Finally, let $\mathcal{S}' = \{S_1, \ldots, S_q\}$, where q is the minimal index for which $\sum_{i=1}^{q} \varphi(S_i) \ge P - P_2$. Note that such an index exists, since $\sum_{i=1}^{k} \varphi(S_i) \ge P_1 - P_2$ and $P_1 > P$. The next lemma bounds the cost of $\mathcal{S}_3 = \mathcal{S}_2 \cup \mathcal{S}'$.

Lemma 4. $c(\mathcal{S}_3) \le c(\mathcal{S}_2) + \alpha c(\mathcal{S}_1 \setminus \mathcal{S}_2) + \epsilon \cdot \mathrm{OPT}$.

Proof. By assumption, the cost of each set in $\mathcal{S}$ is at most $\epsilon \cdot \text{OPT}$. Therefore, it is sufficient to prove that $c(\mathcal{S}' \setminus \{S_q\}) = \sum_{i=1}^{q-1} c(S_i) \le \alpha c(\mathcal{S}_1 \setminus \mathcal{S}_2)$. To this end, consider a random variable K that takes the values $1, \ldots, k$, such that $\mathbb{P}(K = i) = \frac{\varphi(S_i)}{\sum_{l=1}^{k} \varphi(S_l)}$, and let $R = \frac{c(S_K)}{\varphi(S_K)}$. Since the sets in $\mathcal{S}_1 \setminus \mathcal{S}_2$ are indexed by non-decreasing order of $\frac{c(S_i)}{\varphi(S_i)}$, we have $\mathbb{E}(R | 1 \le K \le q-1) \le \mathbb{E}(R)$. As $\alpha = \frac{P - P_2}{P_1 - P_2}$, this inequality implies $\sum_{i=1}^{q-1} c(S_i) \le \alpha c(\mathcal{S}_1 \setminus \mathcal{S}_2)$, since

$$\mathbb{E}(R) = \sum_{i=1}^{k} \frac{c(S_i)}{\varphi(S_i)} \cdot \frac{\varphi(S_i)}{\sum_{l=1}^{k} \varphi(S_l)} = \frac{1}{\sum_{l=1}^{k} \varphi(S_l)} \sum_{i=1}^{k} c(S_i) \le \frac{1}{P_1 - P_2} c(\mathcal{S}_1 \setminus \mathcal{S}_2)$$

and

$$\mathbb{E}(R | 1 \le K \le q-1) = \sum_{i=1}^{q-1} \frac{c(S_i)}{\varphi(S_i)} \cdot \frac{\varphi(S_i)}{\sum_{l=1}^{q-1} \varphi(S_l)} = \frac{1}{\sum_{l=1}^{q-1} \varphi(S_l)} \sum_{i=1}^{q-1} c(S_i) \ge \frac{1}{P - P_2} \sum_{i=1}^{q-1} c(S_i) \ .$$

The last inequality holds since $\sum_{l=1}^{q-1} \varphi(S_l) < P - P_2$, by the minimality of q. ∎

2.4 Deriving the approximation factor

We now conclude the proof of Theorem 1, by demonstrating that the cost of the cheaper of $\mathcal{S}_1$ and $\mathcal{S}_3$ is within factor $(\frac{4}{3} + O(\sqrt{\epsilon}))r$ of optimum. An appropriate choice of ϵ restores the original form of the theorem.

Lemma 5. $\min\{c(\mathcal{S}_1), c(\mathcal{S}_3)\} \le (\frac{4}{3} + O(\sqrt{\epsilon}))r \cdot \text{OPT}$.

Proof. To simplify the analysis, we begin by introducing a new parameter, $\beta = \frac{c(\mathcal{S}_2)}{\text{OPT}} \in [0, r]$, and bound the cost of $\mathcal{S}_1$ and $\mathcal{S}_3$ in terms of OPT, α and β. We first observe that

$$c(\mathcal{S}_1) = \frac{\alpha c(\mathcal{S}_1)}{\alpha} \le \frac{r(1+\epsilon)\text{OPT} - (1-\alpha)c(\mathcal{S}_2)}{\alpha} = \frac{r(1+\epsilon) - (1-\alpha)\beta}{\alpha}\text{OPT} \ ,$$

where the first inequality follows from Lemma 3, and the last equation is obtained by substituting $c(\mathcal{S}_2) = \beta \cdot \text{OPT}$. In addition, Lemma 4 implies that

$$\begin{aligned} c(\mathcal{S}_3) &\le c(\mathcal{S}_2) + \alpha c(\mathcal{S}_1 \setminus \mathcal{S}_2) + \epsilon \cdot \text{OPT} \\ &\le (1-\alpha)c(\mathcal{S}_2) + \alpha c(\mathcal{S}_1) + \alpha c(\mathcal{S}_2) + \epsilon \cdot \text{OPT} \\ &\le r(1+\epsilon)\text{OPT} + \alpha c(\mathcal{S}_2) + \epsilon \cdot \text{OPT} \\ &= (r(1+\epsilon) + \alpha\beta + \epsilon)\text{OPT} \ , \end{aligned}$$

where the third inequality and the last equation follow from Lemma 3 and the definition of β, respectively. Finally, we bound the resulting approximation factor by considering the worst possible choice for the parameters α and β, to conclude that

$$\begin{aligned} \min\{c(\mathcal{S}_1), c(\mathcal{S}_3)\} &\le \min\left\{\frac{r(1+\epsilon) - (1-\alpha)\beta}{\alpha}, r(1+\epsilon) + \alpha\beta + \epsilon\right\}\text{OPT} \\ &\le \max_{\substack{\alpha \in (0,1) \\ \beta \in [0,r]}} \min\left\{\frac{r(1+\epsilon) - (1-\alpha)\beta}{\alpha}, r(1+\epsilon) + \alpha\beta\right\}\text{OPT} + \epsilon \cdot \text{OPT} \\ &= \left(\frac{4}{3} + O(\sqrt{\epsilon})\right) r \cdot \text{OPT} \ . \end{aligned}$$

The last equation is proved in Lemma 6. ∎

Lemma 6.

$$\max_{\substack{\alpha\in(0,1)\\ \beta\in[0,r]}} \min\left\{\frac{r(1+\epsilon)-(1-\alpha)\beta}{\alpha}, r(1+\epsilon)+\alpha\beta\right\} = \left(\frac{4}{3}+O(\sqrt{\epsilon})\right) r \ .$$

Proof. Suppose that $\alpha \le \sqrt{\epsilon}$. In this case, the claim easily follows by observing that for any choice of $\beta \in [0,r]$ we have $r(1+\epsilon)+\alpha\beta \le r(1+\epsilon)+\sqrt{\epsilon}r = (1+O(\sqrt{\epsilon}))r$. We now consider the case $\alpha > \sqrt{\epsilon}$. For fixed $\alpha \in (\sqrt{\epsilon},1)$, let $f_\alpha(\beta) = \frac{r(1+\epsilon)-(1-\alpha)\beta}{\alpha}$ and $g_\alpha(\beta) = r(1+\epsilon)+\alpha\beta$. Note that f_α and g_α are monotone-decreasing and monotone-increasing linear functions of β, respectively. In addition, these functions intersect in the interval $[0,r]$, since $f_\alpha(0) = \frac{(1+\epsilon)r}{\alpha} > (1+\epsilon)r = g_\alpha(0)$ and $f_\alpha(r) = (1+\frac{\epsilon}{\alpha})r < (1+\epsilon+\alpha)r = g_\alpha(r)$, where the middle inequality holds since $\alpha > \sqrt{\epsilon}$. Therefore, $\max_{\beta\in[0,r]} \min\{f_\alpha(\beta), g_\alpha(\beta)\}$ is attained at this intersection point, which is $\beta^* = (1-\frac{\alpha^2}{1-\alpha+\alpha^2})(1+\epsilon)r$, and its value is $f_\alpha(\beta^*) = g_\alpha(\beta^*) = \frac{(1+\epsilon)r}{1-\alpha+\alpha^2}$. The value of α that maximizes the last expression is $\alpha^* = \frac{1}{2}$. It follows that $\beta^* = \frac{2}{3}(1+\epsilon)r$ and

$$\max_{\substack{\alpha\in(\sqrt{\epsilon},1)\\ \beta\in[0,r]}} \min\{f_\alpha(\beta), g_\alpha(\beta)\} = \frac{4}{3}(1+\epsilon)r = \left(\frac{4}{3}+O(\sqrt{\epsilon})\right) r \ .$$

■

3 Applications

In what follows, we demonstrate the applicability of our method on a diverse collection of covering problems, which is by no means exhaustive. Rather, the problems we have chosen to study are only meant to illustrate that the LMP property is applicable in a variety of settings. For the vast majority of these problems, we propose the first algorithm that approximates their generalized partial cover version. For others, our algorithms offer approximation guarantees that compete with the currently best known results.

3.1 Set cover, in terms of Δ

Kearns [26, Thm. 5.15] seems to have been the first to study the partial cover problem, showing that the greedy set cover algorithm [25, 29] can be adapted to provide an approximation factor of $2H(|U|)+3$. A slightly different algorithm was suggested by Slavík [38], who obtained a factor of $H(\min\{\Delta,k\})$, where Δ is the maximum size of a set in $\mathcal{S}$ and k is the coverage requirement. We remark that the partial cover problem contains set cover as a special case, implying that it cannot be approximated within a factor of $(1-\epsilon)\ln|U|$ for any $\epsilon > 0$, unless $\mathrm{NP} \subset \mathrm{TIME}(n^{O(\log\log n)})$ [12].

To the best of our knowledge, the greedy heuristic has not been studied in the context of generalized partial cover, and in fact no algorithm is currently known for this problem. In Subsection 4.1 we prove that every weighted set system $(U,\mathcal{S},c)$ is in $\mathcal{I}_{H(\Delta)}$, where $\Delta = \max_{S\in\mathcal{S}}|S|$. The next theorem follows.

Theorem 7. *The generalized partial set cover problem can be approximated within a factor of* $(\frac{4}{3}+\epsilon)H(\Delta)$*, for any fixed* $\epsilon > 0$.

3.2 Set cover, in terms of f

Let f_e be the number of sets in $\mathcal{S}$ that contain the element $e \in U$; f_e is also known as the *frequency* of e. A recent line of work, that was initiated by Bshouty and Burroughs [5] and Hochbaum [23] in the context of *partial vertex cover*, is approximating partial cover in terms of f, the maximum frequency of any element. Based on the local-ratio method, Bar-Yehuda [3] devised an algorithm for generalized partial cover whose approximation guarantee is f, a result that was independently obtained by Fujito [13] using a prima-dual algorithm. Gandhi, Khuller and Srinivasan [14] achieved a similar ratio for partial cover.

The main result of Subsection 4.2 is a combinatorial f-LMP algorithm for the prize-collecting set cover problem, showing that every weighted set system $(U, \mathcal{S}, c)$ is in $\mathcal{I}_f$, where $f = \max_{e \in U} f_e$. Combined with Theorem 1, this result allows us to approximate the generalized partial set cover problem within a factor of $(\frac{4}{3} + \epsilon)f$, which is slightly worse than the currently best.

3.3 Laminar cover

Let $G = (V, E)$ be an undirected graph, in which each edge $e \in E$ has a non-negative cost $c(e)$, and let $\mathcal{F} = \{V_1, \ldots, V_k\} \subseteq 2^V$ be a *laminar family* of vertex sets, meaning that $V_i \cap V_j \in \{\emptyset, V_i, V_j\}$ for every $i \neq j$. We say that an edge e *covers* V_i if it has exactly one endpoint in V_i. The objective is to find a minimum cost set of edges that collectively cover all sets in $\mathcal{F}$. Note that every instance of this problem induces a weighted set system $(\mathcal{F}, \mathcal{S}, c)$, where for each edge $e \in E$ there is an analogous subset $S_e \in \mathcal{S}$, consisting of all vertex sets $V_i \in \mathcal{F}$ covered by e. Laminar cover can be approximated by applying various techniques, most of which actually deal with the more general *tree augmentation* problem, and produce solutions whose cost is within factor 2 of optimum. We refer the reader to a short survey of these results [11, Sec. 1]. For the unweighted case, Nagamochi [31] proposed a $(1.875 + \epsilon)$-approximation for any fixed $\epsilon > 0$, a ratio that was later improved to $\frac{3}{2}$ by Even, Feldman, Kortsarz and Nutov [11].

In the *generalized partial laminar cover* problem, each $V_i \in \mathcal{F}$ is associated with a profit $p(V_i)$. The goal is to identify a minimum cost set of edges $E' \subseteq E$ such that the overall profit of the sets in $\mathcal{F}$ covered by E' is at least P, a specified profit bound. We are not aware of any approximability result for this problem, even for the seemingly simple case of unit profits. In Subsection 4.3 we prove that $(\mathcal{F}, \mathcal{S}, c) \in \mathcal{I}_2$ for every weighted set system induced by a laminar cover instance, to obtain the following theorem.

Theorem 8. *The generalized partial laminar cover problem can be approximated within a factor of* $\frac{8}{3} + \epsilon$*, for any fixed* $\epsilon > 0$.

3.4 Totally unimodular cover and k-interval cover

The *element-set incidence matrix* $\mathcal{M}_U^{\mathcal{S}}$ of a set system $(U, \mathcal{S})$ has a row for every element $e \in U$ and a column for every set $S \in \mathcal{S}$; its entry in row e and column S is 1 when $e \in S$ and 0 otherwise. Totally unimodular cover (TUC) is a special case of the set cover problem in which $\mathcal{M}_U^{\mathcal{S}}$ is totally unimodular, that is, every square submatrix of this matrix has determinant 0, 1 or -1. We remark that although TUC is known to have integral LP solutions (see, for example, [9, Sec. 6.5]), this property does not extend to its partial covering version, which has not been explicitly studied yet. A particularly interesting problem captured by the latter variant is *partial bipartite vertex cover*: While the approximability of the unit-profit case is still open, arbitrary profits render the problem

NP-hard, since it generalizes *minimum knapsack* even when the given graph is a star. We omit the straightforward reduction.

As illustrated in Subsection 4.2, the prize-collecting set cover problem can be formulated as an integer program whose constraint matrix is $[\mathcal{M}_U^{\mathcal{S}}, I]$. Simple linear algebra arguments show that whenever $\mathcal{M}_U^{\mathcal{S}}$ is totally unimodular then so is $[\mathcal{M}_U^{\mathcal{S}}, I]$, implying that we obtain a 1-LMP algorithm by solving prize-collecting TUC to optimality as a linear program. The next theorem follows.

Theorem 9. *The generalized partial TUC problem can be approximated within a factor of $\frac{4}{3} + \epsilon$, for any fixed $\epsilon > 0$.*

We say that $\mathcal{M}_U^{\mathcal{S}}$ is a *k-interval matrix* if it contains at most k blocks of consecutive 1's in each row. The k-interval cover problem (k-IC) is a special case of set cover in which $\mathcal{M}_U^{\mathcal{S}}$ is a k-interval matrix. In Subsection 4.4 we present a k-LMP rounding algorithm for the prize-collecting k-IC problem, that makes use of our 1-LMP algorithm for the corresponding variant of totally unimodular cover. We derive the following result as a corollary of Theorem 1.

Theorem 10. *The generalized partial k-IC problem can be approximated within a factor of $(\frac{4}{3} + \epsilon)k$, for any fixed $\epsilon > 0$.*

This provides, for instance, the first algorithm that approximates *partial rectangle stabbing* in $\mathbb{R}^d$, noting that the resulting factor of $(\frac{4}{3} + \epsilon)d$ nearly matches the d-approximation of Gaur, Ibaraki and Krishnamurti [19] for the full coverage version of this problem. In addition, we obtain an alternative, albeit non-combinatorial, $(\frac{4}{3} + \epsilon)f$-approximation for partial set cover with maximum element frequency f.

3.5 Edge cover

Given an undirected graph $G = (V, E)$ with non-negative edge costs, *edge cover* is the problem of finding a minimum cost set of edges that contains at least one edge incident to each vertex. Clearly, this problem is equivalent to the special case of set cover in which each subset consists of exactly two elements. We note that edge cover is actually a matching problem in disguise, implying its polynomial time solvability [10, 30]. Plesník [36] proved that unit-profit partial edge cover, which is also known as the *k-edge cover* problem, can be solved to optimality by reducing it to standard edge cover. However, when arbitrary profits are allowed, this problem becomes NP-hard, as it generalizes minimum knapsack. Since Parekh [34, Sec. 2.3] suggested a polynomial-time algorithm for prize-collecting edge cover, we obtain the following theorem.

Theorem 11. *Generalized partial edge cover can be approximated within a factor of $\frac{4}{3} + \epsilon$, for any fixed $\epsilon > 0$.*

3.6 Multicut

On trees. The input to this problem consists of an edge-weighted tree $T = (V, E)$ and a collection of k distinct pairs of vertices, $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$. The objective is to find a minimum cost set of edges whose removal from T disconnects each of the given pairs. It is important to note that, once again, we are facing a special case of set cover: The elements to cover are the input pairs, and an edge $e \in E$ covers $\{s_i, t_i\}$ if it resides on the unique path in T connecting s_i and t_i. Garg, Vazirani and Yannakakis [18] presented a primal-dual 2-approximation for this problem, which was also shown to be at least as hard to approximate as vertex cover.

The corresponding partial cover problem, in which we are required to disconnect a specified number of pairs, has recently been studied by Levin and Segev [28] and independently by Golovin et al. [20], who achieved an approximation guarantee of $\frac{8}{3} + \epsilon$, for any fixed $\epsilon > 0$. Since the former authors provide a 2-LMP algorithm for the prize-collecting multicut problem, we immediately obtain the following theorem, extending the factor of $\frac{8}{3} + \epsilon$ to the case of arbitrary profits.

Theorem 12. *When the underlying graph is a tree, the generalized partial multicut problem can be approximated within a factor of $\frac{8}{3} + \epsilon$, for any fixed $\epsilon > 0$.*

General graphs. When the input graph is no longer restricted to be a tree, the multicut problem becomes significantly harder to approximate. While Garg et al. [17] devised an $O(\log k)$-approximation using the region growing method, a hardness result of $\Omega(\log \log n)$ was given by Chawla, Krauthgamer, Kumar, Rabani and Sivakumar [7], assuming a stronger version of the Unique Games Conjecture [27]. Based on Räcke's hierarchical decomposition method [37], Alon, Awerbuch, Azar, Buchbinder and Naor [1] have shown how to simulate multicuts in general graphs by multicuts in the corresponding decomposition tree. As observed by Golovin et al. [20], this method extends to approximate the partial multicut problem within factor $O(\alpha \log^2 n \log \log n)$, given an α-approximation for the more restricted tree case. Their arguments can be easily combined with Theorem 12 to derive the next result for arbitrary profits.

Theorem 13. *On arbitrary graphs, the generalized partial multicut problem can be approximated within a factor of $O(\log^2 n \log \log n)$.*

4 Prize-Collecting Algorithms

4.1 Every weighted set system is in $\mathcal{I}_{H(\Delta)}$

In the following, we present an $H(\Delta)$-LMP algorithm for all instances of the prize-collecting set cover problem, regardless of any special structure the underlying weighted set system may exhibit. We remind the reader that a prize-collecting instance I_{PC} consists of a ground set U and a family of subsets $\mathcal{S} \subseteq 2^U$, with $\Delta = \max_{S \in \mathcal{S}} |S|$. In addition, the cost of picking a set $S \in \mathcal{S}$ is $c(S)$, and the penalty we incur for leaving an element $e \in U$ uncovered is $\pi(e)$.

The decision not to pick any of the sets that contain an element e, and instead pay its penalty, can be interpreted as picking an implicit singleton $\{e\}$, whose cost is $\pi(e)$. Therefore, we can transform I_{PC} to an instance of the standard set cover problem. However, simple examples demonstrate that a straightforward approach of this nature does not guarantee the LMP property. To this end, rather than setting the cost of each singleton $\{e\}$ to $\pi(e)$, we inflate it by a factor of $H(\Delta)$. As shown in the sequel, this simple adjustment ensures that a greedily constructed solution is indeed $H(\Delta)$-LMP. A formal description of this algorithm is given in Figure 1.

Note that, with respect to I_{PC}, the cost of picking the sets in $\mathcal{S}_{gr}$ is $\sum_{S \in \mathcal{S}_{gr}} c(S)$, whereas the elements left uncovered by $\mathcal{S}_{gr}$ have a total penalty of at most $\sum_{\{e\} \in \mathcal{P}_{gr}} \pi(e)$. We remark that the latter term is not the exact sum of penalties, since $\mathcal{P}_{gr}$ may contain redundant sets.

Lemma 14. $\sum_{S \in \mathcal{S}_{gr}} c(S) + H(\Delta) \sum_{\{e\} \in \mathcal{P}_{gr}} \pi(e) \leq H(\Delta) \cdot \mathrm{OPT}(I_{PC})$.

Proof. Let $\mathcal{S}^* \subseteq \mathcal{S}$ be an optimal solution to I_{PC}, and let $\mathcal{P}^* \subseteq \mathcal{P}$ be the set of singletons $\{e\}$ for which no set in $\mathcal{S}^*$ covers the element e. It is easy to verify that the cost of $\mathcal{S}^*$, as a solution to I_{PC}, is $\sum_{S \in \mathcal{S}^*} c(S) + \sum_{\{e\} \in \mathcal{P}^*} \pi(e) = \mathrm{OPT}(I_{PC})$. At this point, each element $e \in U$ is assigned to a set in $\mathcal{S}^* \cup \mathcal{P}^*$ that covers it, making an arbitrary choice in case of multiple possibilities. Let

1. Construct a set cover instance I_{SC} as follows:
   (a) The ground set is U.
   (b) The family of subsets is $\mathcal{S} \cup \mathcal{P}$, where $\mathcal{P} = \{\{e\} : e \in U\}$.
   (c) The cost of $S \in \mathcal{S}$ is $c(S)$, and the cost of $\{e\} \in \mathcal{P}$ is $H(\Delta)\pi(e)$.
2. Apply the greedy set cover algorithm [25, 29] to obtain an approximate solution for I_{SC}. That is, as long as the sets that were picked so far do not fully cover U, pick a set minimizing the average cost at which it covers new elements.
3. Let $\mathcal{S}_{gr}$ and $\mathcal{P}_{gr}$ be the collections of sets that were picked from $\mathcal{S}$ and $\mathcal{P}$, respectively. Return $\mathcal{S}_{gr}$.

Figure 1: The $H(\Delta)$-LMP prize-collecting algorithm

$\phi : U \to \mathcal{S}^* \cup \mathcal{P}^*$ be the resulting assignment. It is important to note that, by definition of $\mathcal{P}^*$, we have $\phi(e) \in \mathcal{S}^*$ for all elements covered by $\mathcal{S}^*$, and $\phi(e) = \{e\} \in \mathcal{P}^*$ otherwise.

In each iteration of the greedy set cover algorithm, we distribute the cost of the set that had just been picked among the newly covered elements, and denote by price(e) the cost share of e. This charging scheme ensures that the cost of $\mathcal{S}_{gr} \cup \mathcal{P}_{gr}$ with respect to I_{SC} is exactly the sum of cost shares over all elements in U, that is,

$$\sum_{S \in \mathcal{S}_{gr}} c(S) + \sum_{\{e\} \in \mathcal{P}_{gr}} H(\Delta)\pi(e) = \sum_{e \in U} \text{price}(e) \ .$$

Therefore, to complete the proof it is sufficient to show that $\sum_{e \in U} \text{price}(e) \leq H(\Delta) \cdot \text{OPT}(I_{PC})$. We remark that by following the classic analysis, one can bound $\sum_{e \in U} \text{price}(e)$ in terms of $\text{OPT}(I_{SC})$; however, it is quite possible that $\text{OPT}(I_{SC})$ is significantly larger than $\text{OPT}(I_{PC})$, as each singleton $\{e\}$ was given a cost $H(\Delta)\pi(e)$ instead of $\pi(e)$.

Nevertheless, using standard arguments we can prove that $\sum_{e \in \phi^{-1}(S)} \text{price}(e) \leq H(|S|)c(S)$ for every $S \in \mathcal{S}^*$, where $\phi^{-1}(S) = \{e \in U : \phi(e) = S\}$. In addition, price$(e) \leq H(\Delta)\pi(e)$ for every $e \in U$, since the algorithm had the option of picking $\{e\}$ at cost $H(\Delta)\pi(e)$ when the element e was first covered. It follows that

$$\begin{aligned}\sum_{e \in U} \text{price}(e) &= \sum_{\{e\} \notin \mathcal{P}^*} \text{price}(e) + \sum_{\{e\} \in \mathcal{P}^*} \text{price}(e) = \sum_{S \in \mathcal{S}^*} \sum_{e \in \phi^{-1}(S)} \text{price}(e) + \sum_{\{e\} \in \mathcal{P}^*} \text{price}(e) \\ &\leq \sum_{S \in \mathcal{S}^*} H(|S|)c(S) + \sum_{\{e\} \in \mathcal{P}^*} H(\Delta)\pi(e) \leq H(\Delta) \cdot \text{OPT}(I_{PC}) \ .\end{aligned}$$

■

4.2 Every weighted set system is in $\mathcal{I}_f$

In what follows, we propose an f-LMP algorithm for the prize-collecting set cover problem, where f is the maximum frequency of an element in the given set system. We assume that the reader is familiar with the notation introduced in Subsection 4.1, and suggest the following natural LP-relaxation of the problem under consideration:

$$\begin{aligned}
\text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S + \sum_{e \in U} \pi(e) z_e & & \text{(P)} \\
\text{subject to} \quad & \sum_{S \in \mathcal{S}_e} x_S + z_e \geq 1 & \forall\, e \in U & \\
& x_S,\, z_e \geq 0 & \forall\, S \in \mathcal{S},\, e \in U &
\end{aligned}$$

whose dual is given by

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in U} y_e & & \text{(D)} \\
\text{subject to} \quad & \sum_{e \in S} y_e \leq c(S) & \forall\, S \in \mathcal{S} & \\
& 0 \leq y_e \leq \pi(e) & \forall\, e \in U &
\end{aligned}$$

Our method is based on modifying the primal-dual algorithm of Bar-Yehuda and Even [4]. It is not difficult to verify that although the latter algorithm constructs a solution whose cost is at most $(f+1)\mathrm{OPT(P)}$, it does not satisfy the LMP property. For this purpose, we employ an additional elimination phase, in which redundant penalties are discarded. Figure 2 provides a detailed description of the algorithm.

1. Initialize $x = 0$, $z = 0$, $y = 0$.
2. While (x, z) is not a feasible solution to (P),
   (a) Pick an arbitrary element $e \in U$ for which $\sum_{S \in \mathcal{S}_e} x_S + z_e = 0$, and increase the dual variable y_e until $y_e = \pi(e)$ or $\sum_{e' \in S} y_{e'} = c(S)$ for some $S \in \mathcal{S}_e$.
   (b) For every $S \in \mathcal{S}_e$ such that $\sum_{e' \in S} y_{e'} = c(S)$, set $x_S = 1$.
   (c) If $y_e = \pi(e)$, set $z_e = 1$.
3. For every $e \in U$, set $z_e = 0$ if $\sum_{S \in \mathcal{S}_e} x_S \geq 1$.
4. Return (x, z).

Figure 2: The f-LMP prize-collecting algorithm

Lemma 15. *(x, z) is a feasible solution to* (P)*, satisfying* $\sum_{S \in \mathcal{S}} c(S) x_S + f \sum_{e \in U} \pi(e) z_e \leq f \cdot \mathrm{OPT(P)}$.

Proof. We first argue that for every element $e \in U$ either $\sum_{S \in \mathcal{S}_e} x_S \geq 1$ or $z_e = 1$, establishing in particular the feasibility of (x, z). This follows from observing that step 2 guarantees that (x, z) satisfies $\sum_{S \in \mathcal{S}_e} x_S + z_e \geq 1$, whereas step 3 ensures that $\sum_{S \in \mathcal{S}_e} x_S \geq 1$ and $z_e = 1$ do not occur simultaneously.

Now let $\mathcal{P} = \{e \in U : z_e = 1\}$. Note that $y_e = \pi(e)$ for every $e \in \mathcal{P}$, since $z_e = 1$ implies that its corresponding dual constraint $y_e \leq \pi(e)$ is tight. In addition, $c(S) = \sum_{e \in S \setminus \mathcal{P}} y_e$ for all $S \in \mathcal{S}$ with $x_S = 1$. This claim can be easily verified by noting that when $x_S = 1$ we have $c(S) = \sum_{e \in S} y_e$;

however, if $e \in \mathcal{P}$ then $e \notin S$, or otherwise the value of z_e should have been set to 0 in step 3. By combining these observations, we conclude that

$$\sum_{S \in \mathcal{S}} c(S) x_S + f \sum_{e \in U} \pi(e) z_e = \sum_{S \in \mathcal{S}} x_S \sum_{e \in S \setminus \mathcal{P}} y_e + f \sum_{e \in \mathcal{P}} \pi(e) = \sum_{e \notin \mathcal{P}} y_e \sum_{S \in \mathcal{S}_e} x_S + f \sum_{e \in \mathcal{P}} y_e$$
$$\leq \sum_{e \notin \mathcal{P}} y_e |\mathcal{S}_e| + f \sum_{e \in \mathcal{P}} y_e \leq f \sum_{e \notin \mathcal{P}} y_e + f \sum_{e \in \mathcal{P}} y_e = f \sum_{e \in U} y_e \leq f \cdot \text{OPT(P)} \ ,$$

where the last inequality holds since y is a feasible dual solution, and its cost provides a lower bound on OPT(P). ■

4.3 Laminar cover

We now design a 2-LMP algorithm for prize-collecting laminar cover. Recall that an instance I_{PC} of this problem consists of an undirected graph $G = (V, E)$ and a laminar family $\mathcal{F} = \{V_1, \ldots, V_k\} \subseteq 2^V$. In addition, the cost of picking an edge $e \in E$ is $c(e)$, and the penalty for leaving a vertex set $V_i \in \mathcal{F}$ uncovered is $\pi(V_i)$.

A slightly different view. We begin by demonstrating that laminar cover can be transformed into an instance of the *path hitting* problem, that has recently been studied by Parekh and Segev [35]. In the latter, we are given a family of demand paths $\mathcal{D}$ and a family of hitting paths $\mathcal{H}$ in a common undirected tree, where each path $p \in \mathcal{H}$ has a non-negative cost. When $p \in \mathcal{H}$ and $q \in \mathcal{D}$ share at least one mutual edge, we say that *p hits q*. The objective is to find a minimum cost subset of $\mathcal{H}$ whose members collectively hit those of $\mathcal{D}$. A laminar family $\mathcal{F}$ can be represented as a tree $T_{\mathcal{F}}$, in which there is a vertex s_V corresponding to V as well as a vertex s_{V_i} for each $V_i \in \mathcal{F}$. Furthermore, $T_{\mathcal{F}}$ has an edge joining s_{V_i} and s_{V_j} if V_j is the minimal set in $\mathcal{F} \cup \{V\}$ that strictly contains V_i. We assume that this tree is rooted at s_V, and define a set of paths $\mathcal{H}$ as follows. For $u \in V$, let $V[u]$ be the minimal set in $\mathcal{F} \cup \{V\}$ to which u belongs. Then for every $(u, v) \in E$ we add to $\mathcal{H}$ the unique path in $T_{\mathcal{F}}$ connecting $s_{V[u]}$ and $s_{V[v]}$, with cost $c(u, v)$. Using this construction, the laminar cover problem becomes that of finding a minimum cost subset $\mathcal{H}' \subseteq \mathcal{H}$ that hits all edges of $T_{\mathcal{F}}$, which is a special case of path hitting with $\mathcal{D} = E(T_{\mathcal{F}})$. It is important to note that a set V_i is covered if and only if we pick a path that hits the edge connecting s_{V_i} to its parent. Therefore, in the prize-collecting variant the penalty of this edge is identical to that of V_i.

The algorithm. It would be convenient to assume that the instance I_{PC} is already specified in its prize-collecting path hitting representation. Our algorithm creates a new instance I_{PH} of the path hitting problem, in which penalties are incorporated as additional hitting paths with inflated costs. Moreover, we ensure that each resulting path is *descending*, meaning that one of its endpoints is an ancestor of the other. While the path hitting problem is generally NP-hard, Parekh and Segev [35] presented an exact primal-dual algorithm for instances that possess this structural property. A detailed description of the algorithm appears in Figure 3.

Let $\mathcal{E} \subseteq E(T_{\mathcal{F}})$ be the set of edges that are not hit by any path in $\mathcal{H}'$. Note that, with respect to I_{PC}, the cost of picking $\mathcal{H}'$ is $\sum_{p \in \mathcal{H}'} c(p)$, whereas the sum of penalties we incur is exactly $\sum_{e \in \mathcal{E}} \pi(e)$. The following lemma shows that the suggested algorithm is 2-LMP.

Lemma 16. $\sum_{p \in \mathcal{H}'} c(p) + 2 \sum_{e \in \mathcal{E}} \pi(e) \leq 2 \cdot \text{OPT}(I_{PC})$.

Proof. We first observe that $\sum_{p \in \mathcal{H}'} c(p) \leq \sum_{p \in \mathcal{H}^*_S} c(p)$, since each path $p \in \mathcal{H}'$ can be matched to at least one of its replacements in $\mathcal{H}^*_S$, whose cost is identical to that of p. In addition, $\mathcal{E} \subseteq \mathcal{P}^*$, as

1. Construct a path hitting instance I_{PH} as follows:
   (a) The set of demand paths is $E(T_{\mathcal{F}})$.
   (b) The set of hitting paths is $\mathcal{P} \cup \mathcal{H}_S$, where $\mathcal{P} = E(T_{\mathcal{F}})$ and $\mathcal{H}_S$ is created by splitting the paths in $\mathcal{H}$. That is, each $p \in \mathcal{H}$ is replaced by the pair of descending paths that connect the endpoints of p to their lowest common ancestor.
   (c) The cost of $e \in \mathcal{P}$ is $2\pi(e)$, and the cost of each replacement of $p \in \mathcal{H}$ is $c(p)$.
2. Find an optimal solution to I_{PH} using the algorithm of Parekh and Segev [35]. Let $\mathcal{P}^*$ and $\mathcal{H}_S^*$ be the subsets of paths that were picked from $\mathcal{P}$ and $\mathcal{H}_S$, respectively.
3. Let $\mathcal{H}' \subseteq \mathcal{H}$ be the set of original paths p such that at least one of the replacements of p is picked in $\mathcal{H}_S^*$. Return $\mathcal{H}'$.

Figure 3: The 2-LMP prize-collecting algorithm

all edges hit by $\mathcal{H}_S^*$ are also hit by $\mathcal{H}'$. It follows that

$$\sum_{p \in \mathcal{H}'} c(p) + 2 \sum_{e \in \mathcal{E}} \pi(e) \leq \sum_{p \in \mathcal{H}_S^*} c(p) + \sum_{e \in \mathcal{P}^*} 2\pi(e) = \mathrm{OPT}(I_{PH}) \ .$$

To conclude the proof, it is sufficient to show that $\mathrm{OPT}(I_{PH}) \leq 2 \cdot \mathrm{OPT}(I_{PC})$. For this purpose, let $\mathcal{H}^*$ be an optimal solution to I_{PC}, and let $\mathcal{E}^* \subseteq E(T_{\mathcal{F}})$ be the set of edges that are not hit by $\mathcal{H}^*$. We now define a solution to I_{PH} by picking both replacements of each $p \in \mathcal{H}^*$ and all edges in $\mathcal{E}^*$. It can be easily verified that this solution is indeed feasible, and has a total cost of $2 \sum_{p \in \mathcal{H}^*} c(p) + \sum_{e \in \mathcal{E}^*} 2\pi(e) = 2 \cdot \mathrm{OPT}(I_{PC})$ with respect to I_{PH}. ■

4.4 *k*-interval cover

In what follows, we present a k-LMP algorithm for the prize-collecting k-IC problem. We assume that the given instance I_{PC} consists of a ground set U, each element of which is endowed with a penalty $\pi(e)$, and a family of subsets $\mathcal{S} \subseteq 2^U$, where the cost of picking a set $S \in \mathcal{S}$ is $c(S)$. Moreover, the element-set matrix $\mathcal{M}_U^{\mathcal{S}}$ is now a k-interval matrix. In other words, for every element $e \in U$ there is a partition $\{\mathcal{S}_{e,1}, \ldots, \mathcal{S}_{e,k_e}\}$ of $\mathcal{S}_e$ to $k_e \leq k$ subsets, such that the columns corresponding to the sets in each $\mathcal{S}_{e,i}$ appear consecutively in $\mathcal{M}_U^{\mathcal{S}}$.

Our algorithm has its roots in a rounding scheme devised by Gaur et al. [19] for approximating the rectangle stabbing problem. Consider the following LP-relaxation of prize-collecting k-IC, which is a specialization of the one suggested in Subsection 4.2:

$$\begin{aligned}
\text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S + \sum_{e \in U} \pi(e) z_e && \text{(P)} \\
\text{subject to} \quad & \sum_{i=1}^{k_e} \sum_{S \in \mathcal{S}_{e,i}} x_S + z_e \geq 1 && \forall\, e \in U \\
& x_S,\ z_e \geq 0 && \forall\, S \in \mathcal{S},\ e \in U
\end{aligned}$$

We begin by solving the linear program (P) to obtain an optimal fractional solution (x^*, z^*). For every element $e \in U$, let $i(e)$ be the index that maximizes $\sum_{S \in \mathcal{S}_{e,i}} x_S^*$, breaking ties arbitrarily.

Based on these indices, we construct a new program

$$\begin{aligned} \text{minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S + k \sum_{e \in U} \pi(e) z_e & & \qquad (\text{P}^*) \\ \text{subject to} \quad & \sum_{S \in \mathcal{S}_{e,i(e)}} x_S + z_e \geq 1 & \forall\, e \in U & \\ & x_S,\, z_e \geq 0 & \forall\, S \in \mathcal{S},\, e \in U & \end{aligned}$$

Note that the constraint matrix of (P*) can be written as $[\mathcal{M}, I]$, where $\mathcal{M}$ is a matrix that contains a single block of consecutive 1's in each row. Such matrices form a well-known class of totally unimodular matrices (see, for example, [32, page 544]), implying that (P*) is in fact a relaxation of the prize-collecting TUC problem. Therefore, as a result of the discussion in Subsection 3.4, the linear program (P*) has an integral optimal solution $(\hat{x}, \hat{z})$. Since this solution is also feasible to (P), the next lemma shows that the algorithm is indeed k-LMP.

Lemma 17. $\sum_{S \in \mathcal{S}} c(S) \hat{x}_S + k \sum_{e \in U} \pi(e) \hat{z}_e \leq k \cdot \text{OPT(P)}$.

Proof. We first claim that (kx^*, z^*) is a feasible solution to (P*). As the non-negativity constraints are clearly satisfied, it remains to prove that $\sum_{S \in \mathcal{S}_{e,i(e)}} kx^*_S + z^*_e \geq 1$ for every element $e \in U$. To this end, note that since $i(e)$ maximizes $\sum_{S \in \mathcal{S}_{e,i}} x^*_S$, the feasibility of (x^*, z^*) for (P) implies that

$$\sum_{S \in \mathcal{S}_{e,i(e)}} kx^*_S + z^*_e \geq \frac{k}{k_e} \sum_{i=1}^{k_e} \sum_{S \in \mathcal{S}_{e,i}} x^*_S + z^*_e \geq \sum_{i=1}^{k_e} \sum_{S \in \mathcal{S}_{e,i}} x^*_S + z^*_e \geq 1 \ .$$

Now since $(\hat{x}, \hat{z})$ is an optimal solution to (P*), we conclude that

$$\sum_{S \in \mathcal{S}} c(S) \hat{x}_S + k \sum_{e \in U} \pi(e) \hat{z}_e = \text{OPT}(\text{P}^*) \leq \sum_{S \in \mathcal{S}} c(S)(kx^*_S) + k \sum_{e \in U} \pi(e) z^*_e = k \cdot \text{OPT(P)} \ .$$

■

5 Concluding Remarks

Improved approximation factor. Very informally, Theorem 1 states that, given an r-LMP algorithm for the prize-collecting version of a covering problem, we can approximate its partial coverage version to within factor $(\frac{4}{3} + \epsilon)r$ of optimum. However, as mentioned in Section 3, by employing problem-specific techniques a slightly better factor is achievable in some special cases. In light of this observation, it would be interesting to investigate whether the ratio $(\frac{4}{3} + \epsilon)r$ can be improved, perhaps initially just for unit profits.

More than LMP. In an attempt to specialize the framework suggested in this paper to a given application, a seemingly promising idea is to require additional structural properties from the prize-collecting solutions. Based on these properties, the greedy approach we propose in Subsection 2.3 for combining $\mathcal{S}_1$ and $\mathcal{S}_2$ may be replaced by an alternative procedure, resulting in an improved bound on the cost of the final solution.

Implicit subsets. A close inspection of Subsection 2.1 reveals that the main part of our algorithm is repeated $O(|\mathcal{S}|^{1/\epsilon})$ times, once for each guess of the $\lfloor \frac{1}{\epsilon} \rfloor$ most expensive sets in the optimal solution. Therefore, it cannot be directly applied to partial covering problems with a compact implicit representation of exponentially many sets, such as facility location with outliers [6], k-MST [16], k-Steiner forest [21], etc. A challenging open question for future research is whether our method can be extended to approximate problems of this class.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A general approach to online network optimization problems. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 577–586, 2004.

[2] E. Balas and M. Padberg. Set partitioning: A survey. *SIAM Review*, 18(4):710–760, 1976.

[3] R. Bar-Yehuda. Using homogeneous weights for approximating the partial cover problem. *Journal of Algorithms*, 39(2):137–144, 2001.

[4] R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.

[5] N. H. Bshouty and L. Burroughs. Massaging a linear programming solution to give a 2-approximation for a generalization of the vertex cover problem. In *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, pages 298–308, 1998.

[6] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 642–651, 2001.

[7] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 144–153, 2005.

[8] F. A. Chudak, T. Roughgarden, and D. P. Williamson. Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation. *Mathematical Programming*, 100(2):411–421, 2004.

[9] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley and Sons, New York, 1997.

[10] J. Edmonds and E. L. Johnson. Matching: A well-solved class of integer linear programs. In *Combinatorial Structures and their Applications*, pages 89–92. Gordon and Breach, New York, 1970.

[11] G. Even, J. Feldman, G. Kortsarz, and Z. Nutov. A 3/2-approximation algorithm for augmenting the edge-connectivity of a graph from 1 to 2 using a subset of a given edge set. In *Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 90–101, 2001.

[12] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[13] T. Fujito. On approximation of the submodular set cover problem. *Operations Research Letters*, 25(4):169–174, 1999.

[14] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.

[15] R. S. Garfinkel and G. L. Nemhauser. Optimal set covering: A survey. In A. M. Geoffrion, editor, *Perspectives on Optimization*, pages 164–183, 1972.

[16] N. Garg. Saving an epsilon: A 2-approximation for the k-MST problem in graphs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 396–402, 2005.

[17] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

[18] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.

[19] D. R. Gaur, T. Ibaraki, and R. Krishnamurti. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *Journal of Algorithms*, 43(1):138–152, 2002.

[20] D. Golovin, V. Nagarajan, and M. Singh. Approximating the k-multicut problem. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 621–630, 2006.

[21] M. T. Hajiaghayi and K. Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 631–640, 2006.

[22] D. S. Hochbaum. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 3, pages 94–143. PWS Publishing Company, 1997.

[23] D. S. Hochbaum. The t-vertex cover problem: Extending the half integrality framework with budget constraints. In *Proceedings of the 1st International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 111–122, 1998.

[24] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

[25] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.

[26] M. J. Kearns. *The Computational Complexity of Machine Learning*. MIT Press, 1990.

[27] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, 2002.

[28] A. Levin and D. Segev. Partial multicuts in trees. In *Proceedings of the 3rd International Workshop on Approximation and Online Algorithms*, pages 320–333, 2005.

[29] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[30] K. G. Murty and C. Perin. A 1-matching blossom type algorithm for edge covering problems. *Networks*, 12:379–391, 1982.

[31] H. Nagamochi. An approximation for finding a smallest 2-edge-connected subgraph containing a specified spanning tree. *Discrete Applied Mathematics*, 126(1):83–113, 2003.

[32] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, 1988.

[33] M. W. Padberg. Covering, packing and knapsack problems. *Annals of Discrete Mathematics*, 4:265–287, 1979.

[34] O. Parekh. *Polyhedral Techniques for Graphic Covering Problems*. PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University, 2002.

[35] O. Parekh and D. Segev. Path hitting in acyclic graphs. In *Proceedings of the 14th Annual European Symposium on Algorithms*, 2006. To appear.

[36] J. Plesník. Constrained weighted matchings and edge coverings in graphs. *Discrete Applied Mathematics*, 92(2–3):229–241, 1999.

[37] H. Räcke. Minimizing congestion in general networks. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 43–52, 2002.

[38] P. Slavík. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64(5):251–254, 1997.