Covering Problems in Edge- and Node-Weighted Graphs

Takuro Fukunaga*

National Institute of Informatics, Tokyo, Japan
JST, ERATO, Kawarabayashi Large Graph Project, Japan
takuro@nii.ac.jp

Abstract. This paper discusses the graph covering problem in which a set of edges in an edge- and node-weighted graph is chosen to satisfy some covering constraints while minimizing the sum of the weights. In this problem, because of the large integrality gap of a natural linear programming (LP) relaxation, LP rounding algorithms based on the relaxation yield poor performance. Here we propose a stronger LP relaxation for the graph covering problem. The proposed relaxation is applied to designing primal-dual algorithms for two fundamental graph covering problems: the prize-collecting edge dominating set problem and the multicut problem in trees. Our algorithms are an exact polynomial-time algorithm for the former problem, and a 2-approximation algorithm for the latter problem, respectively. These results match the currently known best results for purely edge-weighted graphs.

1 Introduction

1.1 Motivation

Choosing a set of edges in a graph that optimizes some objective function under constraints on the chosen edges constitutes a typical combinatorial optimization problem and has been investigated in many varieties. For example, the spanning tree problem seeks an acyclic edge set that spans all nodes in a graph, the edge cover problem finds an edge set such that each node is incident to at least one edge in the set, and the shortest path problem selects an edge set that connects two specified nodes. All these problems seek to minimize the sum of the weights assigned to edges.

This paper discusses several graph covering problems. Formally, the *graph covering problem* is defined as follows in this paper. Given a graph $G = (V, E)$ and family $\mathcal{E} \subseteq 2^E$, find a subset F of E that satisfies $F \cap C \neq \emptyset$ for each $C \in \mathcal{E}$, while optimizing some function depending on F. As indicated above, the popular approaches assume an edge weight function $w\colon E \to \mathbb{R}_+$ is given, where $\mathbb{R}_+$ denotes the set of non-negative real numbers, and seeks to minimize

* This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Young Scientists (B) 25730008.

R Ravi and I.L. Gørtz (Eds.): SWAT 2014, LNCS 8503, pp. 217–228, 2014.

$\sum_{e \in F} w(e)$. On the other hand, we aspire to simultaneously minimize edge and node weights. Formally, we let $V(F)$ denote the set of end nodes of edges in F. Given a graph $G = (V, E)$ and weight function $w\colon E \cup V \to \mathbb{R}_+$, we seek a subset F of E that minimizes $\sum_{e \in F} w(e) + \sum_{v \in V(F)} w(v)$ under the constraints on F. Hereafter, we denote $\sum_{e \in F} w(e)$ and $\sum_{v \in V(F)} w(v)$ by $w(F)$ and $w(V(F))$, respectively.

Most previous investigations of the graph covering problem have focused on edge weights. By contrast, node weights have been largely neglected, except in the problems of choosing node sets, such as the vertex cover and dominating set problems. To our knowledge, when node weights have been considered in graph covering problems for choosing edge sets, they have been restricted to the Steiner tree problem or its generalizations, possibly because the inclusion of node weights greatly complicates the problem. For example, the Steiner tree problem in edge-weighted graphs can be approximated within a constant factor (the best currently known approximation factor is 1.39 [5,15]). Conversely, the Steiner tree problem in node-weighted graphs is known to extend the set cover problem (see [19]), indicating that achieving an approximation factor of $o(\log |V|)$ is NP-hard. The literature is reviewed in Section 2. As revealed later, the inclusion of node weights generalizes the set cover problem in numerous fundamental problems.

However, from another perspective, node weights can introduce rich structure into the above problems. In fact, node weights provide useful optimization problems. The objective function counts the weight of a node only once, even if the node is shared by multiple edges. Hence, the objective function defined from node weights includes a certain subadditivity, which cannot be captured by edge weights.

The aim of the present paper is to give algorithms for fundamental graph covering problems in edge- and node-weighted graphs. In solving the problems, we adopt a basic linear programming (LP) technique. Algorithms for combinatorial optimization problems are typically designed using LP relaxations. However, in problems with node-weighted graphs, the integrality gap of natural relaxations may be excessively large. Therefore, we propose tighter LP relaxations that preclude unnecessary integrality gaps. We then discuss upper bounds on the integrality gap of these relaxations in two fundamental graph covering problems: the edge dominating set (EDS) problem and multicut problem in trees. We prove upper bounds by designing primal-dual algorithms for both problems. The approximation factors of our proposed algorithms match the current best approximations in purely edge-weighted graphs.

1.2 Problem Definitions

The EDS problem covers edges by choosing adjacent edges in undirected graphs. For any edge e, let $\delta(e)$ denote the set of edges that share end nodes with e, including e itself. We say that an edge e *dominates* another edge f if $f \in \delta(e)$, and a set F of edges dominates an edge f if F contains an edge that dominates f. Given an undirected graph $G = (V, E)$, a set of edges is called an EDS if it dominates each edge in E. The EDS problem seeks to minimize the weight of the EDS. In other words, the EDS problem is the graph covering problem with $\mathcal{E} = \{\delta(e) : e \in E\}$.

The multicut problem specifies an undirected graph $G = (V, E)$ and demand pairs $(s_1, t_1), \ldots, (s_k, t_k) \in V \times V$. A *multicut* is an edge set C whose removal from G disconnects the nodes in each demand pair. This problem seeks a multicut of minimum weight. Let $\mathcal{P}_i$ denote the set of paths connecting s_i and t_i. The multicut problem is equivalent to the graph covering problem with $\mathcal{E} = \bigcup_{i=1}^{k} \mathcal{P}_i$.

Our proposed algorithms for solving these problems assume that the given graph G is a tree. In fact, our algorithms are applicable to the prize-collecting versions of these problems, which additionally specifies a penalty function $\pi : \mathcal{E} \to \mathbb{R}_+$. In this scenario, an edge set F is a feasible solution even if $F \cap C = \emptyset$ for some $C \in \mathcal{E}$, but imposes a penalty $\pi(C)$. The objective is to minimize the sum of $w(F)$, $w(V(F))$, and the penalty $\sum_{C \in \mathcal{E}: F \cap C = \emptyset} \pi(C)$. The prize-collecting versions of the EDS and multicut problems are referred to as the *prize-collecting EDS problem* and the *prize-collecting multicut problem*, respectively.

1.3 Our Results

Thus far, the EDS problem has been applied only to edge-weighted graphs. The vertex cover problem can be reduced to the EDS problem while preserving the approximation factors [6]. The vertex cover problem is solvable by a 2-approximation algorithm, which is widely regarded as the best possible approximation. Indeed, assuming the unique game conjecture, Khot and Regev [18] proved that the vertex cover problem cannot be approximated within a factor better than 2. Fujito and Nagamochi [10] showed that a 2-approximation algorithm is admitted by the EDS problem, which matches the approximation hardness known for the vertex cover problem. In the Appendix, we show that the EDS problem in bipartite graphs generalizes the set cover problem if assigned node weights and generalizes the non-metric facility location problem if assigned edge and node weights. This implies that including node weights increases difficulty of the problem even in bipartite graphs.

On the other hand, Kamiyama [17] proved that the prize-collecting EDS problem in an edge-weighted graph admits an exact polynomial-time algorithm if the graph is a tree. As one of our main results, we show that this idea is extendible to problems in edge- and node-weighted trees.

Theorem 1. *The prize-collecting EDS problem admits a polynomial-time exact algorithm for edge- and node-weighted trees.*

The proof of Theorem 1 will be sketched in Section 4. We can also show that the prize-collecting EDS problem in general edge- and node-weighted graphs admits an $O(\log |V|)$-approximation, which matches the approximation hardness on the set cover problem and the non-metric facility location problem.

The multicut problem is hard even in edge-weighted graphs; the best reported approximation factor is $O(\log k)$ [13]. The multicut problem is known to be both NP-hard and MAX SNP-hard [9], and admits no constant factor approximation algorithm under the unique game conjecture [7]. However, Garg, Vazirani, and Yannakakis [14] developed a 2-approximation algorithm for the multicut problem with edge-weighted trees. They also mentioned that, although the graphs are restricted to trees, the structure of the problem is sufficiently rich. They showed that the tree multicut problem includes the set cover problem with tree-representable set systems. They also showed that the vertex cover problem in general graphs is simply reducible to the multicut problem in star graphs, while preserving the approximation factor. This implies that the 2-approximation seems to be tight for the multicut problem in trees. As a second main result, we extended this 2-approximation to edge- and node-weighted trees, as stated in the following theorem.

Theorem 2. *The prize-collecting multicut problem admits a 2-approximation algorithm for edge- and node-weighted trees.*

Both algorithms claimed in Theorems 1 and 2 are primal-dual algorithms, that use the LP relaxations we propose. These algorithms fall into the same frameworks as those proposed in [14,17] for edge-weighted graphs. However, they need several new ideas to achieve the claimed performance because our LP relaxations are much more complicated than those used in [14,17].

The remainder of this paper is organized as follows. After surveying related work in Section 2, we define our LP relaxation for the prize-collecting graph covering problem in Section 3. In Sections 4, we sketch the proof of Theorem 1 that uses our proposed LP relaxation. The paper concludes with Section 5. We omit the proof of Theorem 2, and discussion on the prize-collecting EDS problem in general graphs with edge- and node-weights, for which we recommend referring to the full version [12] of the current paper.

2 Related Work

As mentioned in Section 1, the graph covering problem in node-weighted graphs has thus far been applied to the Steiner tree problem and its generalizations. Klein and Ravi [19] proposed an $O(\log |V|)$-approximation algorithm for the Steiner tree problem with node weights. Nutov [23,24] extended this algorithm to the survivable network design problem with higher connectivity requirements. An $O(\log |V|)$-approximation algorithm for the prize-collecting Steiner tree problem with node weights was provided by Moss and Rabani [21]; however, as noted by Könemann, Sadeghian, and Sanità [20], the proof of this algorithm contains a technical error. This error was corrected in [20]. Bateni, Hajiaghayi, and Liaghat [1] proposed an $O(\log |V|)$-approximation algorithm for the prize-collecting Steiner forest problem and applied it to the budgeted Steiner tree problem. Chekuri, Ene, and Vakilian [8] gave an $O(k^2 \log |V|)$-approximation algorithm for the prize-collecting survivable network design problem with edge-connectivity requirements of maximum value k. Later, they improved their approximation factor to $O(k \log |V|)$, and also extended it to node-connectivity requirements (see [28]). Naor, Panigrahi, and Singh [22] established an online algorithm for the Steiner tree problem with node weights which was extended to the Steiner forest problem by Hajiaghayi, Liaghat, and Panigrahi [16]. The survivable network design problem with node weights has also been extended to a problem called the network activation problem [26,25,11].

The prize-collecting EDS problem generalizes the $\{0,1\}$-EDS problem, in which given demand edges require being dominated by a solution edge set. The $\{0,1\}$-EDS problem in general edge-weighted graphs admits a $8/3$-approximation, which was proven by Berger et al. [2]. This $8/3$-approximation was extended to the prize-collecting EDS problem by Parekh [27]. Berger and Parekh [3] designed an exact algorithm for the $\{0,1\}$-EDS problem in edge-weighted trees, but their result contains an error [4]. Since the prize-collecting EDS problem embodies the $\{0,1\}$-EDS problem, the latter problem could be alternatively solved by an algorithm developed for the prize-collecting EDS problem in edge-weighted trees, proposed by Kamiyama [17].

3 LP Relaxations

This section discusses LP relaxations for the prize-collecting graph covering problem in edge and node-weighted graphs.

In a natural integer programming (IP) formulation of the graph covering problem, each edge e is associated with a variable $x(e) \in \{0,1\}$, and each node v is associated with a variable $x(v) \in \{0,1\}$. $x(e) = 1$ denotes that e is selected as part of the solution set, while $x(v) = 1$ indicates the selection of an edge incident to v. In the prize-collecting version, each demand set $C \in \mathcal{E}$ is also associated with a variable $z(C) \in \{0,1\}$, where $z(C) = 1$ indicates that the covering constraint corresponding to C is not satisfied. For $F \subseteq E$, we let $\delta_F(v)$ denote the set of edges incident to v in F. The subscript may be removed when $F = E$. An IP of the prize-collecting graph covering problem is then formulated as follows.

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{e \in E} w(e)x(e) + \sum_{v \in V} w(v)x(v) + \sum_{C \in \mathcal{E}} \pi(C)z(C) & \\
\text{subject to} & \displaystyle\sum_{e \in C} x(e) \geq 1 - z(C) & \text{for } C \in \mathcal{E}, \\
& x(v) \geq x(e) & \text{for } v \in V, e \in \delta(v), \\
& x(e) \geq 0 & \text{for } e \in E, \\
& x(v) \geq 0 & \text{for } v \in V, \\
& z(C) \geq 0 & \text{for } C \in \mathcal{E}.
\end{array}$$

In the above formulation, the first constraints specify the covering constraints, while the second constraints indicate that if the solution contains an edge e incident to v, then $x(v) = 1$. In the graph covering problem (without penalties), z is fixed at 0.

To obtain an LP relaxation, we relax the definitions of x and z in the above IP to $x \in \mathbb{R}_+^{E \cup V}$ and $z \in \mathbb{R}_+^{\mathcal{C}}$. However, this relaxation may introduce a large integrality gap into the graph covering problem with node-weighted graphs, as shown in the following example. Suppose that $\mathcal{E}$ comprises a single edge set C, and each edge in C is incident to a node v. Let the weights of all edges and nodes other than v be 0. In this scenario, the optimal value of the graph covering problem is $w(v)$. On the other hand, the LP relaxation admits a feasible solution x such that $x(v) = 1/|C|$ and $x(e) = 1/|C|$ for each edge $e \in C$. The weight of this solution is $w(v)/|C|$, and the integrality gap of the relaxation for this instance is $|C|$. This phenomenon occurs even in the EDS problem and multicut problem in trees.

The above poor example can be excluded if the second constraints in the relaxation are replaced by $x(v) \geq \sum_{e\in\delta(v)} x(e)$ for $v \in V$. However, the LP obtained by this modification does not relax the graph covering problem if the optimal solutions contain high-degree nodes. Thus, we introduce a new variable $y(C, e)$ for each pair of $C \in \mathcal{E}$ and $e \in C$, and replace the second constraints by $x(v) \geq \sum_{e\in\delta(v)} y(C, e)$, where $v \in V$ and $C \in \mathcal{E}$. $y(C, e) = 1$ indicates that e is chosen to satisfy the covering constraint of C, and $y(C, e) = 0$ implies the opposite. Roughly speaking, $y(C, \cdot)$ represents a minimal fractional solution for covering a single demand set C. If a single covering constraint is imposed, the degree of each node is at most one in any minimal integral solution. Then the graph covering problem is relaxed by the LP even after modification. Summing up, we formulate our LP relaxation for an instance $I = (G, \mathcal{E}, w, \pi)$ of the prize-collecting graph covering problem as follows.

$$
\begin{array}{lll}
P(I) = & & \\
\text{minimize} & \displaystyle\sum_{e\in E} w(e)x(e) + \sum_{v\in V} w(v)x(v) + \sum_{C\in\mathcal{E}} \pi(C)z(C) & \\
\text{subject to} & \displaystyle\sum_{e\in C} y(C,e) \geq 1 - z(C) & \text{for } C \in \mathcal{E}, \\
& \displaystyle x(v) \geq \sum_{e\in\delta_C(v)} y(C,e) & \text{for } v \in V, C \in \mathcal{E}, \\
& x(e) \geq y(C,e) & \text{for } C \in \mathcal{E}, e \in C, \\
& x(e) \geq 0 & \text{for } e \in E, \\
& x(v) \geq 0 & \text{for } v \in V, \\
& y(C,e) \geq 0 & \text{for } C \in \mathcal{E}, e \in C, \\
& z(C) \geq 0 & \text{for } C \in \mathcal{E}.
\end{array}
$$

Theorem 3. *Let I be an instance of the prize-collecting graph covering problem in edge- and node-weighted graphs. $P(I)$ is at most the optimal value of I.*

Proof. Let F be an optimal solution of I. We define a solution (x, y, z) of $P(I)$ from F. For each $C \in \mathcal{E}$, we set $z(C)$ to 0 if $F \cap C \neq \emptyset$, and 1 otherwise. If $F \cap C \neq \emptyset$, we choose an arbitrary edge $e \in F \cap C$, and let $y(C, e) = 1$. For the remaining edges e', we assign $y(C, e') = 0$. In this way, the values of variables in y are defined for each $C \in \mathcal{E}$. $x(e)$ is set to 1 if $e \in F$, and 0 otherwise. $x(v)$ is set to 1 if F contains an edge incident to v, and 0 otherwise. (x, y, z) is feasible, and its objective value in $P(I)$ is the optimal value of I. □

In some graph covering problems, $\mathcal{E}$ is not explicitly given, and $|\mathcal{E}|$ is not bounded by a polynomial on the input size of the problem. In such cases, the above LP may not be solved in polynomial time because it cannot be written compactly. However, in this scenario, we may define a tighter LP than the natural relaxation if we can find $\mathcal{E}_1, \ldots, \mathcal{E}_t \subseteq \mathcal{E}$ such that $\cup_{i=1}^{t} \mathcal{E}_i = \mathcal{E}$, t is bounded by a polynomial of input size, and the degree of each node is small in any minimal edge set covering all demand sets in $\mathcal{E}_i$ for each $i \in \{1, \ldots, t\}$. Applying these conditions, the present author obtained a new approximation algorithm for solving a problem generalizing some prize-collecting graph covering problems [11].

4 Prize-Collecting EDS Problem in Trees

In this section, we prove Theorem 1. We regard the input graph G as a rooted tree, with an arbitrary node r selected as the root. The *depth* of a node v is the number of edges on the path between r and v. When v lies on the path between r and another node u, we say that v is an *ancestor* of u and u is a *descendant* of v. If the depth of node v is the maximum among all ancestors of u, then v is defined as the *parent* of u. If v is the parent of u, then u is a *child* of v. The upper and lower end nodes of an edge e are denoted by u_e and l_e, respectively. We say that an edge e is an ancestor of a node v and v is a descendant of e when $l_e = v$ or l_e is an ancestor of v. Similarly, an edge e is a descendant of a node v and v is an ancestor of e if $v = u_e$ or v is an ancestor of u_e. An edge e is defined as an ancestor of another edge f if e is an ancestor of u_f.

Recall that $\mathcal{E} = \{\delta(e) \colon e \in E\}$ in the EDS problem. Let $I = (G, w, \pi)$ be an instance of the prize-collecting EDS problem. We denote $\bigcup_{e \in \delta(v)} \delta(e)$ by $\delta'(v)$ for each $v \in V$. Then the dual of $P(I)$ is formulated as follows.

$$
\begin{array}{llll}
D(I) = & & & \\
\text{maximize} & \displaystyle\sum_{e \in E} \xi(e) & & \\
\text{subject to} & \displaystyle\sum_{e \in \delta(e')} \nu(e', e) \le w(e') & \text{for } e' \in E, & (1) \\
& \displaystyle\sum_{e \in \delta'(v)} \mu(v, e) \le w(v) & \text{for } v \in V, & (2) \\
& \xi(e) \le \mu(u, e) + \mu(v, e) + \nu(e', e) & \text{for } e \in E, e' = uv \in \delta(e), & (3) \\
& \xi(e) \le \pi(e) & \text{for } e \in E, & (4) \\
& \xi(e) \ge 0 & \text{for } e \in E, & \\
& \nu(e', e) \ge 0 & \text{for } e' \in E, e \in \delta(e'), & \\
& \mu(v, e) \ge 0 & \text{for } v \in V, e \in \delta'(v). &
\end{array}
$$

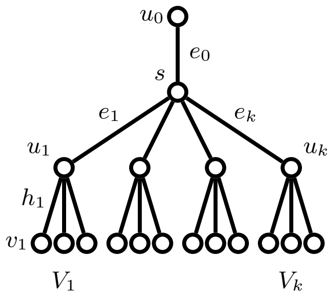

Fig. 1. Edges and nodes in Case B

For an edge set $F \subseteq E$, let $\tilde{F}$ denote $\{e \in E \colon \delta_F(e) = \emptyset\}$, and let $\pi(\tilde{F})$ denote $\sum_{e \in \tilde{F}} \pi(e)$. For the instance I, our algorithm yields a solution $F \subseteq E$ and a feasible solution (ξ, ν, μ) to $D(I)$, both satisfying

$$w(F) + w(V(F)) + \pi(\tilde{F}) \leq \sum_{e \in E} \xi(e). \tag{5}$$

Since the right-hand side of (5) is at most $P(I)$, F is an optimal solution of I. We note that the dual solution (ξ, ν, μ) is required only for proving the optimality of the solution and need not be computed.

The algorithm operates by induction on the number of nodes of depth exceeding one. In the base case, all nodes are of depth one, indicating that G is a star centered at r. The alternative case is divided into two sub-cases: Case A, in which a leaf edge e of maximum depth satisfies $\pi(e) > 0$; and Case B, which contains no such leaf edge. In this paper, we discuss only Case B due to the space limination.

Case B

In this case, $\pi(e) = 0$ holds for all leaf edges e of maximum depth. Let s be the grandparent of a leaf node of maximum depth. Also, let $u_1, \ldots, u_k$ be the children of s, and e_i be the edge joining s and u_i for $i \in [k]$. In the following discussion, we assume that s has a parent, and that each node u_i has at least one child. This discussion is easily modified to cases in which s has no parent or some node u_i has no child. We denote the parent of s by u_0, and the edge between u_0 and s by e_0. For each $i \in [k]$, let V_i be the set of children of u_i, and H_i be the set of edges joining u_i to its child nodes in V_i. Also define $h_i = u_i v_i$ as an edge that attains $\min_{u_i v \in H_i}(w(u_i v) + w(v))$. The relationships between these nodes and edges are illustrated in Fig. 1.

Now define $\theta_1 = \min_{i=0}^{k}(w(e_i) + w(u_i) + w(s))$, $\theta_2 = \sum_{i=1}^{k} \min\{w(u_i) + w(v_i) + w(h_i), \pi(e_i)\}$, and let $\theta = \min\{\theta_1, \theta_2\}$. We denote the index $i \in [k]$ of an edge e_i that attains $\theta_1 = w(e_i) + w(u_i) + w(s)$ by i^*, and specify $K = \{i \in [k] \colon w(u_i) + w(v_i) + w(h_i) \leq \pi(e_i)\}$. For a real number ψ, we let $(\psi)_+$ denote $\max\{0, \psi\}$.

We define $I' = (G', w', \pi')$ as follows. If $\theta_1 \geq \theta_2$, then G' is the tree obtained by removing all edges in $\bigcup_{i \in [k]} H_i$ and all nodes in $\bigcup_{i \in [k]} V_i$ from G, and $\pi' \colon E' \to \mathbb{R}_+$ is defined such that

$$\pi'(e) = \begin{cases} 0 & \text{if } e \in \{e_1, \ldots, e_k\}, \\ \pi(e) & \text{otherwise} \end{cases}$$

for $e \in E'$. In this case, $w' \colon V' \cup E' \to \mathbb{R}_+$ is defined by

$$w'(v) = \begin{cases} (w(s) - \theta)_+ & \text{if } v = s, \\ w(u_i) - (\theta - w(s) - w(e_i))_+ & \text{if } v = u_i, i \in [k]^* \\ w(v) & \text{otherwise} \end{cases}$$

for $v \in V'$, and

$$w'(e) = \begin{cases} (w(e_i) - (\theta - w(s))_+)_+ & \text{if } e = e_i, i \in [k]^*, \\ w(e) & \text{otherwise,} \end{cases}$$

for $e \in E'$. If $\theta_1 < \theta_2$, then $e_1, \ldots, e_k$, and their descendants are removed from G to obtain G', and π' is defined by

$$\pi'(e) = \begin{cases} 0 & \text{if } e = e_0, \\ \pi(e) & \text{otherwise.} \end{cases}$$

Moreover, w' for E' and V' is defined as in the case $\theta_1 \geq \theta_2$, disregarding the weights of edges and nodes removed from G'.

Since G' has fewer nodes of depth exceeding one than G, the algorithm inductively finds a solution F' to I', and a feasible solution (ξ', ν', μ') to $D(I')$ satisfying (5). F is constructed from F' as follows.

$$F = \begin{cases} F' \cup \{e_0\} & \text{if } \delta_{F'}(u_0) \neq \emptyset, \theta > w(s) + w(e_0), \\ F' & \text{if } \delta_{F'}(u_0) = \emptyset \text{ or } \theta \leq w(s) + w(e_0), \delta_{F'}(s) \neq \emptyset, \\ F' \cup \{h_i \colon i \in K\} & \text{if } \delta_{F'}(u_0) = \emptyset \text{ or } \theta \leq w(s) + w(e_0), \delta_{F'}(s) = \emptyset, \theta_1 \geq \theta_2, \\ F' \cup \{e_{i^*}\} & \text{if } \delta_{F'}(u_0) = \emptyset \text{ or } \theta \leq w(s) + w(e_0), \delta_{F'}(s) = \emptyset, \theta_1 < \theta_2. \end{cases}$$

We define $\xi(e_1), \ldots, \xi(e_k)$ such that $\xi(e_i) \leq \min\{w(u_i) + w(v_i) + w(h_i), \pi(e_i)\}$ for $i \in [k]$ and $\sum_{i=1}^k \xi(e_i) = \theta$, which is possible because $\sum_{i=1}^k \min\{w(u_i) + w(v_i) + w(h_i), \pi(e_i)\} = \theta_2 \geq \theta$. We also define $\xi(e) = 0$ for each $e \in \bigcup_{i=1}^k H_i$. The other variables in ξ are set to their values in ξ'. The following lemma states that this ξ can form a feasible solution to $D(I)$.

Lemma 1. *Suppose that $\xi(e_1), \ldots, \xi(e_k)$ satisfy $\xi(e_i) \leq \min\{w(u_i) + w(v_i) + w(h_i), \pi(e_i)\}$ for each $i \in [k]$ and $\sum_{i=1}^k \xi(e_i) = \theta$. Further, suppose that $\xi(e) = 0$ holds for each $e \in \bigcup_{i=1}^k H_i$, and the other variables in ξ are set to their values in ξ'. Then there exist ν and μ such that (ξ, ν, μ) is feasible to $D(I)$.*

Proof. For $i \in [k]$ and $v \in V_i$, we define $\mu(v, e_i)$ and $\nu(u_i v, e_i)$ such that $\mu(v, e_i) + \nu(u_i v, e_i) = \min\{w(v_i) + w(h_i), \xi(e_i)\}$. This may be achieved without violating the constraints, because $w(v) + w(u_i v) \geq w(v_i) + w(h_i)$. We also define $\nu(u_i v, e_i)$ as $(\xi(e_i) - w(u_i) - w(h_i))_+$. These variables satisfy (1) for $u_i v$, (2) for v and u_i, and (3) for $(e_i, u_i v)$. $\nu(e_j, e_i)$ for $i \in [k]$ and $j \in [k]^*$, and $\mu(v, e_i)$ for $i \in [k]$ and $v \in \{s\} \cup \{u_j \colon j \in [k]^*, j \neq i\}$ are set to 0. The other variables in ν and μ are set to their values in ν' and μ'. To advance the proof, we introduce an algorithm that increases $\nu(e_j, e_i)$ for $i \in [k]$ and $j \in [k]^*$, and $\mu(v, e_i)$ for $i \in [k]$ and $v \in \{s, u_0, \ldots, u_k\}$. At the completion of the algorithm, (ξ, ν, μ) is a feasible solution to $D(I)$.

The algorithm performs k iterations, and the i-th iteration increases the variables to satisfy (3) for each pair of e_i and e_j, where $j \in [k]^*$. The algorithm retains a set `Var` of variables to be increased. We introduce a notion of time: Over one unit of time, the algorithm simultaneously increases all variables in `Var` by one. The time consumed by the i-th iteration is $\xi(e_i)$.

At the beginning of the i-th iteration, `Var` is initialized to $\{\mu(u_j, e_i) \colon j \in [k]^*\}$. The algorithm updates `Var` during the i-th iteration as follows.

- At time $(\xi(e_i) - w(v_i) - w(h_i))_+$, $\mu(u_i, e_i)$ is added to `Var` if `Var` $\neq \{\mu(s, e_i)\}$;
- If (2) becomes tight for u_j under the increase of $\mu(u_j, e_i) \in$ `Var`, then $\mu(u_j, e_i)$ is replaced by $\nu(e_j, e_i)$ for each $j \in [k]^*$;
- If (1) becomes tight for e_j under the increase of $\nu(e_j, e_i) \in$ `Var` with some $j \in [k]^*$, then `Var` is reset to $\{\mu(s, e_i)\}$.

We note that the time spent between two consecutive updates may be zero.

`Var` always contains a variable that appears in the right-hand side of (3) for (e_i, e_j) with $j \in [k]^* \setminus \{i\}$, and for (e_i, e_i) after time $(\xi(e_i) - w(v_i) - w(h_i))_+$. The algorithm updates `Var` so that (1) and (2) hold for all variables except s. Hence, to show that (ξ, ν, μ) is a feasible solution to $D(I)$, it suffices to show that (2) for s does not become tight before the algorithm is completed.

We complete the proof by contradiction. Suppose that (2) for s tightens at time $\tau < \xi(e_i)$ in the i-th iteration. Since `Var` $= \{\mu(s, e_i)\}$ at this moment, there exists $j \in [k]^*$ such that (1) for e_j and (2) for u_j are tight. The variables in the left-hand sides of (1) for e_j and (2) for u_j and s are not simultaneously increased. Nor are these variables increased over time $(\xi(e_j) - w(v_j) - w(h_j))_+$ in the j-th iteration, and $\mu(u_j, e_j)$ is initialized to $(\xi(e_j) - w(v_j) - w(h_j))_+$. From this argument, it follows that $w(s) + w(u_j) + w(e_j) < \sum_{i'=1}^{k} \xi(e_{i'}) \leq \theta$. However, this result is contradicted by the definition of θ, which implies that $\theta \leq \theta_1 \leq w(s) + w(u_j) + w(e_j)$. Thus, the claim is proven. □

Lemma 2. *F and ξ satisfy* (5).

Proof. For each $i \in [k]$, either $e_i \notin E'$ holds, or $\xi'(e_i) = 0$ holds (because $\pi'(e_i) = 0$). Hence, $\sum_{e \in E} \xi(e) = \sum_{i=1}^{k} \xi(e_i) + \sum_{e \in E'} \xi'(e) = \theta + \sum_{e \in E'} \xi'(e)$. Therefore, it suffices to prove that $\sum_{e \in F} w(e) \leq \theta + \sum_{e \in F'} w'(e)$.

Without loss of generality, we can assume $|\delta_{F'}(e_0)| \leq 1$ (if false, we can remove edges e_i, $i \in [k]^*$ from F' until $|\delta_{F'}(e_0)| = 1$). In the sequel, we discuss only the case of $\delta_{F'}(u_0) \neq \emptyset$ and $\theta > w(s) + w(e_0)$. In the alternative case, the claim immediately follows from the definitions of F and w'. $\delta_{F'}(u_0) \neq \emptyset$ implies that $w'(u_0)$ is counted in the objective value of F'. Moreover, $w'(s) = w'(e_0) = 0$ follows from $\theta > w(s) + w(e_0)$. Thus, the objective values increase from F' to F by $w(u_0) - w'(u_0) + w(e_0) + w(s)$, which equals θ. □

5 Conclusion

In this paper, we emphasized a large integrality gap when the natural LP relaxation is applied to the graph covering problem that minimizes node weights. We then formulated an alternative LP relaxation for graph covering problems in edge- and node-weighted graphs that is stronger than the natural relaxation. This relaxation was incorporated into an exact algorithm for the prize-collecting EDS problem in trees, and a 2-approximation algorithm for the multicut problem in trees. The approximation guarantees for these algorithms match the previously known best results for purely edge-weighted graphs. In many other graph covering problems, the integrality gap in the proposed relaxation would increase if node weights were introduced, because the problems in node-weighted graphs admit stronger hardness results. Nonetheless, the proposed relaxation is a potentially useful tool for designing heuristics or using IP solvers to solve the above problems.

References

1. Bateni, M., Hajiaghayi, M., Liaghat, V.: Improved approximation algorithms for (budgeted) node-weighted Steiner problems. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) ICALP 2013, Part I. LNCS, vol. 7965, pp. 81–92. Springer, Heidelberg (2013)
2. Berger, A., Fukunaga, T., Nagamochi, H., Parekh, O.: Approximability of the capacitated b-edge dominating set problem. Theoretical Computer Science 385(1-3), 202–213 (2007)
3. Berger, A., Parekh, O.: Linear time algorithms for generalized edge dominating set problems. Algorithmica 50(2), 244–254 (2008)
4. Berger, A., Parekh, O.: Erratum to: Linear time algorithms for generalized edge dominating set problems. Algorithmica 62(1-2), 633–634 (2012)
5. Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L.: Steiner tree approximation via iterative randomized rounding. Journal of the ACM 60(1), 6 (2013)
6. Carr, R.D., Fujito, T., Konjevod, G., Parekh, O.: A $2\frac{1}{10}$-approximation algorithm for a generalization of the weighted edge-dominating set problem. Journal of Combinatorial Optimization 5(3), 317–326 (2001)
7. Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., Sivakumar, D.: On the hardness of approximating multicut and sparsest-cut. Computational Complexity 15(2), 94–114 (2006)
8. Chekuri, C., Ene, A., Vakilian, A.: Prize-collecting survivable network design in node-weighted graphs. In: Gupta, A., Jansen, K., Rolim, J., Servedio, R. (eds.) APPROX/RANDOM 2012. LNCS, vol. 7408, pp. 98–109. Springer, Heidelberg (2012)

9. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM Journal on Computing 23(4), 864–894 (1994)
10. Fujito, T., Nagamochi, H.: A 2-approximation algorithm for the minimum weight edge dominating set problem. Discrete Applied Mathematics 118(3), 199–207 (2002)
11. Fukunaga, T.: Spider covers for prize-collecting network activation problem. CoRR abs/1310.5422 (2013)
12. Fukunaga, T.: Covering problems in edge- and node-weighted graphs. CoRR abs/1404.4123 (2014)
13. Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi)cut theorems and their applications. SIAM Journal on Computing 25(2), 235–251 (1996)
14. Garg, N., Vazirani, V.V., Yannakakis, M.: Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica 18(1), 3–20 (1997)
15. Goemans, M.X., Olver, N., Rothvoß, T., Zenklusen, R.: Matroids and integrality gaps for hypergraphic steiner tree relaxations. In: STOC, pp. 1161–1176 (2012)
16. Hajiaghayi, M., Liaghat, V., Panigrahi, D.: Online node-weighted Steiner forest and extensions via disk paintings. In: FOCS, pp. 558–567 (2013)
17. Kamiyama, N.: The prize-collecting edge dominating set problem in trees. In: Hliněný, P., Kučera, A. (eds.) MFCS 2010. LNCS, vol. 6281, pp. 465–476. Springer, Heidelberg (2010)
18. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within $2-\epsilon$. Journal of Computer and System Sciences 74(3), 335–349 (2008)
19. Klein, P.N., Ravi, R.: A nearly best-possible approximation algorithm for node-weighted Steiner trees. Journal of Algorithms 19(1), 104–115 (1995)
20. Könemann, J., Sadeghabad, S.S., Sanità, L.: An LMP $O(\log n)$-approximation algorithm for node weighted prize collecting Steiner tree. In: FOCS, pp. 568–577 (2013)
21. Moss, A., Rabani, Y.: Approximation algorithms for constrained node weighted Steiner tree problems. SIAM Journal on Computing 37(2), 460–481 (2007)
22. Naor, J., Panigrahi, D., Singh, M.: Online node-weighted Steiner tree and related problems. In: FOCS, pp. 210–219 (2011)
23. Nutov, Z.: Approximating Steiner networks with node-weights. SIAM Journal on Computing 39(7), 3001–3022 (2010)
24. Nutov, Z.: Approximating minimum-cost connectivity problems via uncrossable bifamilies. ACM Transactions on Algorithms 9(1), 1 (2012)
25. Nutov, Z.: Survivable network activation problems. In: Fernández-Baca, D. (ed.) LATIN 2012. LNCS, vol. 7256, pp. 594–605. Springer, Heidelberg (2012)
26. Panigrahi, D.: Survivable network design problems in wireless networks. In: SODA, pp. 1014–1027 (2011)
27. Parekh, O.: Approximation algorithms for partially covering with edges. Theoretical Computer Science 400(1-3), 159–168 (2008)
28. Vakilian, A.: Node-weighted prize-collecting survivable network design problems. Master's thesis, University of Illinois at Urbana-Champaign (2013)