The Prize-Collecting Edge Dominating Set Problem in Trees

Naoyuki Kamiyama*

Department of Information and System Engineering, Chuo University
kamiyama@ise.chuo-u.ac.jp

Abstract. In this paper, we consider the prize-collecting edge dominating set problem, which is a generalization of the edge dominating set problem. In the prize-collecting edge dominating set problem, we are not forced to dominate all edges, but we need to pay penalties for edges which are not dominated. It is known that this problem is $\mathcal{NP}$-hard, and Parekh presented a $\frac{8}{3}$-approximation algorithm. To the best of our knowledge, no polynomial-time solvable case is known for this problem. In this paper, we show that the prize-collecting edge dominating set problem in trees can be solved in polynomial time.

1 Introduction

Throughout this paper, we denote by $\mathbb{Z}_+$ and $\mathbb{R}_+$ the sets of nonnegative integers and nonnegative real numbers, respectively. Given a function or a vector f on a ground set U, we use the notation $f(X) = \sum_{e \in X} f(e)$ for each $X \subseteq U$.

Let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E. In this paper, we regard an edge as a set of exactly two vertices. For each $X \subseteq V$, we denote by $\delta(X)$ be the set of $e \in E$ such that $e \cap X \neq \emptyset$. For each $v \in V$, we use the notation $\delta(v)$ instead of $\delta(\{v\})$. We say that $F \subseteq E$ *dominates* $e \in E$ if $e \in \delta(f)$ for some $f \in F$. We call $F \subseteq E$ an *edge dominating set of* G if F dominates all the edges of E. The *edge dominating set problem* asks for finding an edge dominating set of G with minimum cardinality. This problem is one of fundamental covering problems such as the vertex cover problem.

Yannakakis and Gavril [1] proved that the edge dominating set problem is $\mathcal{NP}$-hard in a graph which is planar or bipartite of maximum degree 3. Gotthilf, Lewenstein and Rainshmidt [2] presented a $(2 - c\frac{\log n}{n})$-approximation algorithm which is based on the local search technique, where c is an arbitrary constant and n is the number of vertices. As a special case, several classes of graph in which this problem can be solved in polynomial time are known, e.g., trees [3].

In the *weighted edge dominating set problem*, we are given a weight function $w \colon E \to \mathbb{R}_+$, and this problem asks for finding an edge dominating set of G with minimum weight, where the weight of $F \subseteq E$ is defined by $w(F)$. Fujito and Nagamochi [4] and Parekh [5] independently presented a 2-approximation

* Supported by Grants-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

P. Hliněný and A. Kučera (Eds.): MFCS 2010, LNCS 6281, pp. 465–476, 2010.

algorithm for this problem. As a special case, Berger and Parekh [6] gave a polynomial-time algorithm for this problem in trees.

In this paper, we consider the *prize-collecting edge dominating set problem*, which is a generalization of the weighted edge dominating set problem. Recall that in the edge dominating set problem we have to dominate *all* edges. However, in the prize-collecting edge dominating set problem we are not forced to dominate all edges, but we need to pay *penalties* for edges which are not dominated. More formally, the prize-collecting edge dominating set problem is defined as follows. We are given a graph $G = (V, E)$, a weight function $w\colon E \to \mathbb{R}_+$ and a penalty function $\pi\colon E \to \mathbb{R}_+$. The *cost* of $F \subseteq E$ is defined by $w(F) + \pi(F')$, where F' denotes the set of the edges of E which are not dominated by F. The prize-collecting edge dominating set problem asks for finding a subset of E with minimum cost. Prize-collecting type variants of combinatorial optimization problems have been extensively studied (for example, see [7,8]).

For the prize-collecting edge dominating set problem, Parekh [9] gave a $\frac{8}{3}$-approximation algorithm. To the best of our knowledge, no polynomial-time solvable case is known for this problem. In this paper, we show that the prize-collecting edge dominating set problem in trees can be solved in polynomial time. Our algorithm is based on the algorithm of Berger and Parekh [6] for the weighted edge dominating set problem in trees, but we should emphasize that the extension is non-trivial.

The rest of this paper is organized as follows. Section 2 gives our algorithm for the prize-collecting edge dominating set problem in trees. In Section 3, we show the correctness of our algorithm. In Section 4, we consider the time complexity of our algorithm. In Section 5, we consider the total dual integrality of a polyhedron related to our problem and some generalization.

Notations. Let $G = (V, E)$ be a tree, and we specify an arbitrary vertex of G as a root. For each $v \in V$, the *depth* of v is the number of the edges contained in the (unique) path from the root to v (denoted by $d(v)$), and the *parent* of v is the (unique) vertex $u \in V$ such that $d(u) = d(v) - 1$ and $\{u, v\} \in E$. For each $v \in V$, let pv and e_v be the parent of v and the edge $\{v, pv\}$, respectively. We say that $v \in V$ is a *child* of pv. For each $v \in V$, we denote by gv the parent of pv.

2 Algorithm

In this section, we present an algorithm for the prize-collecting edge dominating set problem in trees. More precisely, we give an algorithm for a more general problem, called the *prize-collecting b-edge dominating set problem*. In this problem, we are given a graph $G = (V, E)$, a weight function $w\colon E \to \mathbb{R}_+$, a penalty function $\pi\colon E \to \mathbb{R}_+$ and a demand function $b\colon E \to \{0, 1\}$. The cost of $F \subseteq E$ is defined by $w(F) + \pi(\overline{F})$, where $\overline{F}$ is the set of $e \in E$ such that $b(e) = 1$ and e is not dominated by F. If $b(e) = 1$ for all $e \in E$, the prize-collecting b-edge dominating set problem is equivalent to the prize-collecting edge dominating set problem.

In the sequel, let G be a tree with a root. For each $v \in V$, let $\delta_1(v)$ be the set of $e \in \delta(v)$ such that $b(e) = 1$, let $\delta^c(v)$ be the set of $e \in E$ between v and its children, and let $\delta^c_1(v)$ (resp., $\delta^c_0(v)$) be the set of $e \in \delta^c(v)$ such that $b(e) = 1$ (resp., $b(e) = 0$). We denote by M the set of $v \in V$ such that $d(v) = \max_{u \in V} d(u)$, and let M_1 be the set of $v \in M$ such that $b(e_v) = 1$.

Our algorithm is recursively defined according to the number of vertices $v \in V$ such that $d(v) > 1$. For the base case, we consider the case where there exists no $v \in V$ such that $d(v) > 1$, i.e., G is a star. Let E_1 be the set of $e \in E$ such that $b(e) = 1$. If $\min_{e \in E} w(e) \leq \pi(E_1)$, the algorithm outputs a minimizer of $\min_{e \in E} w(e)$. Otherwise, the algorithm outputs $\emptyset$.

From here we consider the case where there exists $v \in V$ such that $d(v) > 1$. We divide this case into the following subcases.

Case A: $M_1 \neq \emptyset$.
Case A1: There exists $v \in M_1$ such that $b(e_{pv}) = 0$.
Case A2: There exists $v \in M_1$ such that $b(e_{pv}) = 1$ and

$$\min_{e \in \delta(pv)} w(e) \leq \sum_{e \in \delta^c_1(pv)} \pi(e). \tag{1}$$

Case A3: For all $v \in M_1$ such that $b(e_{pv}) = 1$, (1) does not hold.
Case B: $M_1 = \emptyset$.

In the rest of this section, we give the detail of our algorithm for each case. Given a subgraph $G' = (V', E')$ of G, a weight function $w' \colon E' \to \mathbb{R}_+$ and a demand function $b' \colon E' \to \{0, 1\}$, an *instance* $I' = (G', w', b')$ is the prize-collecting b'-edge dominating set problem in G' with w' and the restriction of π on E'. Namely, the original problem is an instance $I = (G, w, b)$. Let AI be the output of our algorithm for the instance I. (We adopt this notation for any instance I', i.e., we denote by AI' the output of our algorithm to I'.)

2.1 Case A1

In this subsection, we give an algorithm for Case A1. Let $v \in M_1$ be a vertex such that $b(e_{pv}) = 0$. We define α by

$$\alpha = \min\Big(\min_{e \in \delta(pv)} w(e), \sum_{e \in \delta^c_1(pv)} \pi(e)\Big).$$

Let $G' = (V', E')$ be the graph obtained from G by removing the children of pv and the edges of $\delta^c(pv)$. We define a weight function w' on E' by

$$w'(e) = \begin{cases} w(e) - \alpha, & \text{if } e = e_{pv}, \\ w(e), & \text{otherwise.} \end{cases}$$

We define a demand function b' on E' by $b'(e) = b(e)$ for all edges of E'. Let I' be an instance (G', w', b'). If $e_{pv} \in AI'$, our algorithm outputs AI', i.e., $AI = AI'$. If $e_{pv} \notin AI'$ and

$$\min_{e \in \delta(pv)} w(e) \leq \sum_{e \in \delta^c_1(pv)} \pi(e), \tag{2}$$

the algorithm outputs $AI' \cup \{e^*\}$, where e^* is a minimizer of the left-hand side of (2). If $e_{pv} \notin AI'$ and (2) does not hold, our algorithm outputs AI'.

2.2 Case A2

In this subsection, we give an algorithm for Case A2. Let $v \in M_1$ be a vertex such that $b(e_{pv}) = 1$ and (1) holds. Let $G' = (V', E')$ be the graph obtained from G by removing the children of pv and the edges of $\delta^c(pv)$. We define a weight function w' on E' by

$$w'(e) = \begin{cases} w(e) - \min_{e \in \delta(pv)} w(e), & \text{if } e = e_{pv}, \\ w(e), & \text{otherwise.} \end{cases}$$

We define a demand function b' on E' by $b'(e_{pv}) = 0$ and $b'(e) = b(e)$ for the other edges of E'. Let I' be an instance (G', w', b'). If $e_{pv} \in AI'$, our algorithm outputs AI'. Otherwise, the algorithm outputs $AI' \cup \{e^*\}$, where e^* is a minimizer of the left-hand side of (1).

2.3 Case A3

In this subsection, we give an algorithm for Case A3. Let $v \in M_1$ be a vertex such that $b(e_{pv}) = 1$ and (1) does not hold. We define β and γ by

$$\beta = \min\Big\{ \min_{e \in \delta(pv)} w(e), \sum_{e \in \delta_1(pv)} \pi(e) \Big\} - \sum_{e \in \delta_1^c(pv)} \pi(e),$$

$$\gamma = \min\Big(\min_{e \in \delta(gv) \setminus \{e_{pv}\}} w(e), \beta \Big).$$

Let $G' = (V', E')$ be the graph obtained from G by removing the children of pv and the edges of $\delta^c(pv)$. We define a weight function w' on E' by

$$w'(e) = \begin{cases} w(e) - \Big(\gamma + \sum_{e \in \delta_1^c(pv)} \pi(e)\Big), & \text{if } e = e_{pv}, \\ w(e) - \gamma, & \text{if } e \in \delta(gv) \setminus \{e_{pv}\}, \\ w(e), & \text{otherwise.} \end{cases}$$

We define a demand function b' on E' by $b'(e_{pv}) = 0$ and $b'(e) = b(e)$ for the other edges of E'. Let I' be an instance (G', w', b'). If AI' contains an edge of $\delta(gv)$, our algorithm outputs AI'. From here, we consider the case where AI' does not contain an edge of $\delta(gv)$. If

$$\min_{e \in \delta(gv) \setminus \{e_{pv}\}} w(e) \le \beta, \tag{3}$$

our algorithm outputs $AI' \cup \{e_1^*\}$, where e_1^* is a minimizer of the left-hand side of (3). If (3) does not hold and

$$\min_{e \in \delta(pv)} w(e) \le \sum_{e \in \delta_1(pv)} \pi(e), \tag{4}$$

our algorithm outputs $AI' \cup \{e_2^*\}$, where e_2^* is a minimizer of the left-hand side of (4). If (3) and (4) do not hold, our algorithm outputs AI'.

2.4 Case B

In this subsection, we give an algorithm for Case B. Let v be a vertex of M. Without loss of generality we can assume that $b(e_{pv}) = 1$. Otherwise, we can remove edges of $\delta^c(pv)$. We define η by

$$\eta = \min\Big(\min_{e\in\delta(e_{pv})} w(e), \pi(e_{pv})\Big)$$

Let $G' = (V', E')$ be the graph obtained from G by removing the children of pv and the edges of $\delta^c(pv)$. We define a weight function w' on E' by

$$w'(e) = \begin{cases} w(e) - \eta, & \text{if } e \in \delta(gv), \\ w(e), & \text{otherwise.} \end{cases}$$

We define a demand function b' on E' by $b'(e_{pv}) = 0$ and $b'(e) = b(e)$ for the other edges of E'. Let I' be an instance (G', w', b'). If AI' contain an edge of $\delta(gv)$, our algorithm outputs AI'. If AI' does not contain an edge of $\delta(gv)$ and

$$\min_{e\in\delta(e_{pv})} w(e) \leq \pi(e_{pv}), \tag{5}$$

our algorithm outputs $AI' \cup \{e^*\}$, where e^* is a minimizer of the left-hand side of (5). If AI' does not contains an edge of $\delta(e_{pv})$ and (5) does not hold, our algorithm outputs AI'.

3 Correctness

In this section, we prove the correctness of the algorithm presented in Section 2. The integer programming formulation IP(I) of an instance $I = (G, w, b)$ can be described as follows.

$$\text{IP}(I) \left|\begin{array}{ll} \min & \langle w, x\rangle + \langle \pi, y\rangle \\ \text{s.t.} & x(\delta(e)) + y(e) \geq b(e) \quad (\forall e \in E) \\ & x, y \in \mathbb{Z}_+^E, \end{array}\right.$$

where let $\langle f, g\rangle$ be the inner product of f and g for vectors f and g on the same ground set. The dual problem Dual(I) of the linear programming relaxation of IP(I) can be described as follows.

$$\textsc{Dual}(I) \left|\begin{array}{ll} \max & \langle b, z\rangle \\ \text{s.t.} & z(e) \leq \pi(e) \quad (\forall e \in E) \\ & z(\delta(e)) \leq w(e) \quad (\forall e \in E) \\ & z \in \mathbb{R}_+^E. \end{array}\right.$$

By the weak duality theorem [10], a feasible solution z to Dual(I) such that $w(AI) + \pi(\overline{AI}) \leq \langle b, z\rangle$, AI is optimal to an instance I. Hence, in order to show the correctness, it suffices to prove the following lemma.

Lemma 1. *Given an instance $I = (G, w, b)$, there exists a feasible solution z to* $\textsc{Dual}(I)$ *such that $w(AI) + \pi(\overline{AI}) \leq \langle b, z\rangle$.*

In the sequel, we show Lemma 1 for each case by induction on the number of $v \in V$ such that $d(v) > 1$. For an instance I, we denote by $c(I)$ the cost of a solution which our algorithm outputs, i.e., $c(I) = w(AI) + \pi(\overline{AI})$.

3.1 The Star Case

In this subsection, we consider the case where G is a star. We first consider the case of $\min_{e \in E} w(e) \leq \pi(E_1)$. We arbitrarily name the edges of E_1 as $e_1, \ldots, e_m$, where $m = |E_1|$. Since $\min_{e \in E} w(e) \leq \pi(E_1)$, there exists the index $l \in \{1, \ldots, m\}$ such that

$$\sum_{i=1}^{l-1} \pi(e_i) < \min_{e \in E} w(e) \leq \sum_{i=1}^{l} \pi(e_i).$$

We define $z \in \mathbb{R}_+^E$ as follows. Set $z(e_i) = \pi(e_i)$ for all $i \in \{1, \ldots, l-1\}$, and

$$z(e_l) = \min_{e \in E} w(e) - \sum_{i=1}^{l-1} \pi(e_i).$$

Set $z(e) = 0$ for the other edges e. By the definition, $z(e) \leq \pi(e)$ and $z(\delta(e)) = \min_{f \in E} w(f) \leq w(e)$ for all $e \in E$. Hence, z is feasible for $\textsc{Dual}(I)$. Furthermore, $c(I) = \min_{e \in E} w(e)$ and $\langle b, z\rangle = \min_{e \in E} w(e)$ clearly holds.

Next we consider the case of $\min_{e \in E} w(e) > \pi(E_1)$. In this case, set $z(e) = \pi(e)$ for all $e \in E_1$ and $z(e) = 0$ for the other edges e. By the definition, $z(e) \leq \pi(e)$ for all $e \in E$. Since $\min_{e \in E} w(e) > \pi(E_1)$, $z(\delta(e)) = \pi(E_1) < w(e)$ for all $e \in E$. Hence, z is feasible to $\textsc{Dual}(I)$. Furthermore, $c(I) = \pi(E_1)$ and $\langle b, z\rangle = \pi(E_1)$ clearly holds. This completes the proof.

3.2 Case A1

By the induction hypothesis, there exists a feasible solution z' to $\textsc{Dual}(I')$ such that $c(I') \leq \langle b', z'\rangle$. We define $z \in \mathbb{R}_+^E$ as follows. For each $e \in \delta_1^c(pv)$, define $z(e)$ so that $z(e) \leq \pi(e)$ and

$$\sum_{f \in \delta_1^c(pv)} z(f) = \alpha.$$

By the definition α, we can do this in the same manner for the star case. For each $e \in \delta_0^c(pv)$, set $z(e) = 0$. For the other edges e of E', define $z(e) = z'(e)$.

Now we consider the feasibility of z. By the definition, $z(e) \leq \pi(e)$ for all $e \in \delta^c(pv)$. Hence, by the induction hypothesis, $z(e) \leq \pi(e)$ for all $e \in E$. Next we consider the second condition. Here it should note that we can assume that

$z'(e_{pv}) = 0$ since $b'(e_{pv}) = 0$ holds. Since $z(e_{pv}) = 0$ holds by $z(e_{pv}) = z'(e_{pv})$, $z(\delta(e)) = \alpha \le w(e)$ for each $e \in \delta^c(pv)$. By the definition of w',

$$z(\delta(e_{pv})) = z'(\delta(gv)) + \alpha \le w(e_{pv}).$$

For the other edges, the condition is satisfied by the induction hypothesis, which implies the feasibility of z.

Finally, we show that $c(I) \le \langle b, z\rangle$. In every case, $c(I) - c(I') \le \alpha$ and $\langle b, z\rangle - \langle b', z'\rangle = \alpha$. This completes the proof.

3.3 Case A2

By the induction hypothesis, there exists a feasible solution z' to $\textsc{Dual}(I')$ such that $c(I') \le \langle b', z'\rangle$. We define $z \in \mathbb{R}^E_+$ as follows. For each $e \in \delta^c_1(pv)$, define $z(e)$ so that $z(e) \le \pi(e)$ and

$$\sum_{f \in \delta^c_1(pv)} z(f) = \min_{f \in \delta(pv)} w(f).$$

Since (1) holds, we can do this in the same manner for the star case. For each $e \in \delta^c_0(pv)$, set $z(e) = 0$. For the other edges e of E', define $z(e) = z'(e)$.

Now we consider the feasibility of z. By the definition, $z(e) \le \pi(e)$ for all $e \in \delta^c(pv)$. Hence, by the induction hypothesis, $z(e) \le \pi(e)$ for all $e \in E$. Next we consider the second condition. We can assume that $z'(e_{pv}) = 0$ since $b'(e_{pv}) = 0$ holds. Since $z(e_{pv}) = 0$ holds by $z(e_{pv}) = z'(e_{pv})$,

$$z(\delta(e)) = \sum_{f \in \delta^c_1(pv)} z(f) = \min_{f \in \delta(pv)} w(f) \le w(e)$$

for each $e \in \delta^c(pv)$. By the definition of w',

$$z(\delta(e_{pv})) = z'(\delta(gv)) + \min_{e \in \delta(pv)} w(e) \le w(e_{pv}).$$

For the other edges, the condition is satisfied by the induction hypothesis, which implies the feasibility of z.

Finally, we show that $c(I) \le \langle b, z\rangle$. In both cases,

$$c(I) - c(I') \le \min_{e \in \delta(pv)} w(e), \quad \langle b, z\rangle - \langle b', z'\rangle = \min_{e \in \delta(pv)} w(e).$$

This completes the proof.

3.4 Case A3

By the induction hypothesis, there exists a feasible solution z' to $\textsc{Dual}(I')$ such that $c(I') \le \langle b', z'\rangle$. We define $z \in \mathbb{R}^E_+$ by

$$z(e) = \begin{cases} \pi(e), & \text{if } e \in \delta^c_1(v), \\ 0, & \text{if } e \in \delta^c_0(v), \\ \gamma, & \text{if } e = e_{pv}, \\ z'(e), & \text{otherwise.} \end{cases}$$

First we consider the feasibility of z. By the definition, $z(e) \leq \pi(e)$ for all $e \in \delta^c(pv)$. By the definition of γ, $z(e_{pv}) = \gamma \leq \pi(e_{pv})$ holds. Hence, by the induction hypothesis, $z(e) \leq \pi(e)$ for all $e \in E$. Next we consider the second condition. For each $e \in \delta^c(pv)$.

$$z(\delta(e)) = \gamma + \sum_{e \in \delta_1^c(pv)} \pi(e) \leq \min_{f \in \delta(pv)} w(f) \leq w(e). \tag{6}$$

Furthermore, by the definition of w',

$$z(\delta(e_{pv})) = z'(\delta(gv)) + \gamma + \sum_{e \in \delta_1^c(pv)} \pi(e) \leq w(e_{pv}). \tag{7}$$

For each $e \in \delta(gv) \setminus \{e_{pv}\}$,

$$z(\delta(e)) = z'(\delta(e)) + \gamma \leq w(e) \tag{8}$$

Notice that in (6)-(8) the first equation follows from that we can assume $z'(e_{pv}) = 0$ since $b'(e_{pv}) = 0$. For the other edges, the condition is satisfied by the induction hypothesis, which implies the feasibility of z.

Next we show that $c(I) \leq \langle b, z \rangle$. For this, we first show that we can assume that AI' contains at most one edge of $\delta(gv)$. Suppose that AI' contains more than one edge of $\delta(gv)$. In this case, AI' contains at least one edge e of $\delta^c(gv)$. When e is incident to a leaf, $c(I')$ does not increase by removing e from AI'. If e is not incident to a leaf, $c(I')$ does not increase removing e from AI' since (2) does not hold. Hence, by the optimality of AI', we can assume that AI' contains at most one edge of $\delta(gv)$. By this fact, in every case $c(I) - c(I')$ is at most

$$\gamma + \sum_{e \in \delta_1^c(pv)} \pi(e), \tag{9}$$

by $z'(e_{pv}) = 0$ and $\langle b, z \rangle - \langle b', z' \rangle$ is equal to (9). This completes the proof.

3.5 Case B

By the induction hypothesis, there exists a feasible solution z' to $\textsc{Dual}(I')$ such that $c(I') \leq \langle b', z' \rangle$. We define $z \in \mathbb{R}_+^E$ as follows.

$$z(e) = \begin{cases} 0, & \text{if } e \in \delta^c(v), \\ \eta, & \text{if } e = e_{pv}, \\ z'(e), & \text{otherwise.} \end{cases}$$

First we consider the feasibility of z. By the definition of η, $z(e_{pv}) = \eta \leq \pi(e_{pv})$ holds. Hence, by the induction hypothesis, $z(e) \leq \pi(e)$ for all $e \in E$. Next we consider the second condition. For each $e \in \delta^c(pv)$, $z(\delta(e)) = \eta \leq w(e)$. By the definition of w',

$$z(\delta(e)) = z'(\delta(e)) + \eta \leq w(e)$$

for each $e \in \delta(gv) \setminus \{e_{pv}\}$. The first equation follows from that we can assume $z'(e_{pv}) = 0$ since $b'(e_{pv}) = 0$ holds. Furthermore, by the definition of w',

$$z(\delta(e_{pv})) = z'(\delta(gv)) + \eta \le w(e_{pv}).$$

For the other edges, the condition is satisfied by the induction hypothesis, which implies the feasibility of z.

Next we show that $c(I) \le \langle b, z \rangle$. For this, we first show that we can assume that AI' contains at most one edge of $\delta(gv)$. Suppose that AI' contains more than one edge of $\delta(gv)$. In this case, AI' contains at least one edge e of $\delta^c(gv)$. Since $b(e_v) = 0$ for all $v \in M$, $c(I')$ does not increase by removing e from AI'. Hence, by the optimality of AI', we can assume that AI' contains at most one edge of $\delta(gv)$. By this fact, in the both cases $c(I) - c(I') \le \eta$ and $\langle b, z \rangle - \langle b', z' \rangle = \eta$. This completes the proof.

4 Time Complexity

In this section, we consider the time complexity of our algorithm. Our algorithm, called **Algorithm PEDS**, can be described as follows.

Algorithm PEDS

1. Compute $d(v)$ for all $v \in V$, and set $l = \max_{v \in V} d(v)$.
2. If $l > 1$, remove all the vertices $v \in V$ such that $d(v) = l$, and change the weight function and the demand function in the manner describe in Section 2, whiling keeping the followings in mind.
   (a) We give priority to vertices for which the conditions of Case A1, Case A2, Case A3 and Case B in this order.
   (b) When we remove a vertex for which the conditions of Case A3, some vertices may become to satisfy the conditions of Case A2, i.e., (1). In this case, we remove these vertices before other vertices for which the conditions of Case A3.
   (c) After removing all the vertices $v \in V$ such that $d(v) = l$, update $l = l - 1$.
3. Notice that in this step the input graph becomes a star. Hence, compute an optimal solution to the star, and construction an optimal solution to the original problem.

Theorem 1. *Given a tree $G = (V, E)$, a weight function $w\colon E \to \mathbb{R}_+$, a penalty function $\pi\colon E \to \mathbb{R}_+$ and a demand function $b\colon E \to \{0, 1\}$,* **Algorithm PEDS** *can solve the prize-collecting b-edge dominating set problem in $O(|V|^2)$ time.*

Proof. Since the correctness of our algorithm is proved in the previous section, we consider the time complexity. Clearly, we can complete Step 1 in $O(|V|)$ time. Also Step 3 can be done in $O(|V|)$ time by storing the order in which vertices are removed. Hence, the time required to complete Step 2 is dominating factor of our algorithm. Although the problem is that we have to check some vertex become to satisfy the conditions of Case A2, we can do this in constant time by storing a difference of the both sides of (1). Hence, we can do Step 4 in $O(|V_l|^2)$ time for each l, where V_l is the set of $v \in V$ such that $d(v) = l$. This completes the proof. □

Here we give a bad example for which our algorithm requires $\Omega(|V|^2)$ time. A vertex set V consists of a root r and vertices $X = \{x_1, \ldots, x_k\}$ and $Y = \{y_1, \ldots, y_k\}$. An edge set E consists of $\{r, x_i\}$ and $\{x_i, y_i\}$ for all $i \in \{1, \ldots, k\}$. For all $e \in E$, we define $b(e) = 1$, and $w(e) = +\infty$ and $\pi(e) = 1$. Then, whenever we remove a vertex of Y, we have to update the weights of all the edges of $\delta(r)$. Hence, our algorithm requires $\Omega(|V|^2)$ time for this instance.

5 Total Dual Integrality and Generalization

In this section, we consider the total dual integrality of a polytope related to our problem and some generalization.

5.1 Total Dual Integrality of a Related Polyhedron

In our algorithm, we have an *integral* dual optimal solution if a weight and a penalty of each edge are integral. Hence, we can obtain the following polyhedral result. Let A be a $p \times q$-matrix in which every entry is rational, and let b be a p-dimensional rational vector. Then, a system $Ax \geq b$ for $x \in \mathbb{R}^q_+$ is called *total dual integral* when for each $l \in \mathbb{Z}^q$ the dual problem of minimizing $\langle l, x \rangle$ over $Ax \geq b$ has an integer optimal solution if the dual problem has a feasible solution and the optimal objective value of the dual problem is finite. It is known [11] that if a system $Ax \geq b$ for $x \in \mathbb{R}^p_+$ is total dual integral and $b \in \mathbb{Z}^p$, every vertex of the polyhedron which is determined by a system $Ax \geq b$ has integer coordinates.

Theorem 2. *Given a tree $G = (V, E)$ and a demand function $b \colon E \to \{0, 1\}$, a system $\{x(\delta(e)) + y(e) \geq b(e) \mid e \in E\}$ for $x, y \in \mathbb{R}^E_+$ is total dual integral.*

Proof. For a weight function $w \colon E \to \mathbb{Z}_+$ and a penalty function $\pi \colon E \to \mathbb{Z}_+$, it follows from the proof of the correctness of our algorithm that $\textsc{Dual}(I)$ has an integral optimal solution, where I is an instance (G, w, b). For a weight function w and a penalty function π such that there exists $e \in E$ such that $w(e) < 0$ or $\pi(e) < 0$, $\textsc{Dual}(I)$ has no solution. This completes the proof. □

5.2 Generalization

Here we consider a generalization of the prize collecting b-edge dominating set problem. It is natural to generalize an image of a demand function b from $\{0, 1\}$ to $\mathbb{Z}_+$. More precisely, the *generalized prize-collecting b-edge dominating set problem* is defined as follows. We are given a graph $G = (V, E)$, a weight function $w \colon E \to \mathbb{R}_+$, a penalty function $\pi \colon E \to \mathbb{R}_+$ and a demand function $b \colon E \to \mathbb{Z}_+$. The cost of a vector $x \in \mathbb{Z}^E_+$ is defined by $\langle w, x \rangle + \langle \pi, \overline{x} \rangle$, where $\overline{x} \in \mathbb{R}^E_+$ is defined $\max\{0, b(e) - x(\delta(e))\}$ for each $e \in E$. Notice that the integer programming formulation of this problem is also IP(I). Although it is open whether the generalized prize-collecting b-edge dominating set problem in trees can be solved in polynomial time, by using the theory of *total unimodular* we can show that this problem in a path G can be solved in polynomial time. A matrix A is called totally unimodular when if every square submatrix has determinant $0, \pm 1$. The following theorem is known (see also [10, Corollary 19.2a]).

Theorem 3 (Hoffman and Kruskal [12]). *Let A be a totally unimodular $p \times q$-matrix. Then, for each $b \in \mathbb{Z}^p$, every extreme point of the polyhedron determined by a system $Ax \geq b$ for $x \in \mathbb{R}_+$ has integer coordinates.*

We define the *edge-edge adjacency matrix* A_G of G as follows. Letting $|E| = m$ and $E = \{e_1, \ldots, e_m\}$, A_G is an $m \times m$-matrix whose entry corresponding to a i-th row and a j-th column is defined by 1 if $e_i \cap e_j \neq \emptyset$, and 0 otherwise. By Theorem 3, if $[A_G, \Delta]$ is totally unimodular, the generalized prize-collecting b-edge dominating set problem can be solved in polynomial time by solving the linear programming relaxation of IP(I), where Δ is an identity matrix and $[A_G, \Delta]$ is a matrix obtained by combining A_G and Δ. If G is a path, we can prove this by using the following theorems.

Theorem 4 (Schrijver [10, Example 7 in p.279]). *If every entry of a matrix A is 0 or 1 and each row of A has its 1's consecutively, A is totally unimodular.*

Theorem 5 (Ghoulia-Houri [13]). *A matrix A is totally unimodular if and only if each collection R of rows of A can be partitioned into classes R_1 and R_2 such that the sum of the rows in R_1 minus the sum of the rows in R_2 is a vector with entries $0, \pm 1$ only.*

If G is a path, it follows from Theorem 4 that A_G is totally unimodular, and then each collection R of rows of A_G can be partitioned into classes R_1 and R_2 satisfying the condition in Theorem 5. It is clear that for the classes R_1 and R_2, Δ satisfies the condition in Theorem 5. Hence, it follows from Theorem 5 that $[A_G, \Delta]$ is totally unimodular, which implies the polynomial-time solvability of the generalized prize-collecting b-edge dominating set problem in paths.

Theorem 6. *The generalized prize-collecting b-edge dominating set problem in paths can be solved in polynomial time.*

References

1. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. SIAM Journal on Applied Mathematics 38(3), 364–372 (1980)
2. Gotthilf, Z., Lewenstein, M., Rainshmidt, E.: A $(2 - c\frac{\log n}{n})$ approximation algorithm for the minimum maximal matching problem. In: Bampis, E., Skutella, M. (eds.) WAOA 2008. LNCS, vol. 5426, pp. 267–278. Springer, Heidelberg (2009)
3. Mitchell, S., Hedetniemi, S.: Edge domination in trees. In: Proceedings of the Eighth Southern Conference on Combinatorics, Graph Theory, and Computing, pp. 489–509 (1977)

4. Fujito, T., Nagamochi, H.: A 2-approximation algorithm for the minimum weight edge dominating set problem. Discrete Applied Mathematics 118(3), 199–207 (2002)
5. Parekh, O.: Edge dominating and hypomatchable sets. In: Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002), pp. 287–291 (2002)
6. Berger, A., Parekh, O.: Linear time algorithms for generalized edge dominating set problems. Algorithmica 50(2), 244–254 (2008)
7. Archer, A., Bateni, M., Hajiaghayi, M.T., Karloff, H.J.: Improved approximation algorithms for prize-collecting steiner tree and TSP. In: Proceedings of the Fiftieth Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009), pp. 427–436 (2009)
8. Hochbaum, D.S.: Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. European Journal of Operational Research 140(2), 291–321 (2002)
9. Parekh, O.: Approximation algorithms for partially covering with edges. Theoretical Computer Science 400(1-3), 159–168 (2008)
10. Schrijver, A.: Theory of Linear and Integer Programming. J. Wiley & Sons, Chichester (1986)
11. Edmonds, J., Giles, R.: A min–max relation for submodular functions on graphs. Annals of Discrete Mathematics 1, 185–204 (1977)
12. Hoffman, A.J., Kruskal, J.B.: Integral boundary points of convex polyhedra. In: Kuhn, H.W., Tucker, A.W. (eds.) Linear Inequalities and Related Systems, pp. 223–246. Princeton University Press, Princeton (1956)
13. Ghouila-Houri, A.: Caractérisation des matrices totalement unimodulaires. Comptes Redus Hebdomadaires des Séances de l'Académie des Sciences (Paris) 254, 1192–1194 (1962)