

# Linear Time Algorithms for Generalized Edge Dominating Set Problems

André Berger · Ojas Parekh

Received: 1 January 2006 / Accepted: 23 May 2007 / Published online: 6 October 2007  
© Springer Science+Business Media, LLC 2007

**Abstract** We prove that a generalization of the edge dominating set problem, in which each edge  $e$  needs to be covered  $b_e$  times for all  $e \in E$ , admits a linear time 2-approximation for general unweighted graphs and that it can be solved optimally for weighted trees. We show how to solve it optimally in linear time for unweighted trees and for weighted trees in which  $b_e \in \{0, 1\}$  for all  $e \in E$ . Moreover, we show that it solves generalizations of weighted matching, vertex cover, and edge cover in trees.

**Keywords** Edge dominating set · Approximation algorithm · Trees

## 1 Introduction

Domination problems in graphs have been subject of many studies in graph theory, and have many applications in operations research, e.g., in resource allocation and network routing, as well as in coding theory.

In this paper we consider a generalization of the *edge dominating set (EDS) problem*. Given a graph  $G = (V, E)$ , a function  $b : E \rightarrow \mathbb{N}$  and a weight function  $c : E \rightarrow \mathbb{Q}^+$ , a  $b$ -EDS is a subset  $F \subseteq E$  together with a multiplicity  $m_e \in \mathbb{N}^+$  for each  $e \in F$ , so that each edge in  $E$  is adjacent to at least  $b_e = b(e)$  edges in  $F$ , counting multiplicities. The  $b$ -EDS problem is then to find a  $b$ -EDS which minimizes  $\sum_{e \in F} m_e$  in the unweighted and  $\sum_{e \in F} (c(e) \cdot m_e)$  in the weighted case. The  $b$ -EDS

---

A. Berger (✉)

Department of Mathematics, Technical University Berlin, 10623 Berlin, Germany  
e-mail: berger.andre@gmail.com

O. Parekh

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

problem generalizes the EDS problem in much the same way that the set multicover problem generalizes the set cover problem [19].

When  $b_e = 1$  for all  $e \in E$  this is the edge dominating set problem (EDS). It is the least studied problem among the four natural covering problems in graphs: edge cover (cover  $V$  with elements from  $E$ ), vertex cover ( $E$  with  $V$ ), dominating set ( $V$  with  $V$ ), and EDS ( $E$  with  $E$ ).

Our motivation for studying EDS is that it is a fundamental covering problem in graphs which simultaneously generalizes and melds vertex cover and edge cover into a restricted version of the total cover problem [2, 14]. Moreover the weighted matching problem may be solved as an instance of weighted edge cover, hence a striking feature of the linear time primal-dual algorithm which we present is that it solves generalizations of weighted matching, vertex cover, and edge cover in trees.

The unweighted version of EDS is NP-complete even for planar and bipartite graphs of maximum degree 3 [20] as well as for several other families of graphs [9]. However, there are also families of graphs for which the unweighted EDS problem is polynomial-time solvable [9, 18]. In particular, linear time algorithms for the unweighted version are known for trees [11] and block graphs [10].

Much less is known about the weighted version of the problem. Recently, Fujito and Nagamochi [5] and Parekh [13] independently discovered a 2-approximation for the weighted EDS problem; the latter also showed that weighted EDS restricted to bipartite graphs is no easier to approximate than weighted vertex cover, which is MAX-SNP-hard [12] and is suspected to have no polynomial time approximation algorithm with approximation ratio asymptotically less than 2. Thus, a 2-approximation for  $b$ -EDS may be the best we can hope for. Indeed, in [1] it is shown that the  $b$ -EDS problem can be approximated within a factor of  $8/3$ , but also that this factor cannot be improved using the same methods that were used in [5] and [13].

When  $b_e \in \{0, 1\}$  for all  $e \in E$  we call the resulting problem  $\{0, 1\}$ -EDS. The weighted version of  $\{0, 1\}$ -EDS is particularly interesting since it is equivalent to the generalization of weighted vertex cover in which in addition to single vertices, weights may also be assigned to pairs,  $\{u, v\}$ , of vertices (by adding an edge  $uv$  with  $b_{uv} = 0$  if one does not already exist). This generalization may be used to model a limited economy of scale in existing applications of vertex cover: for a pair of vertices  $\{u, v\}$  one may stipulate that selecting both  $u$  and  $v$  costs less than the sum of the individual costs of  $u$  and  $v$ .

Our main contributions are linear-time algorithms for three special cases of the  $b$ -EDS problem, a 2-approximation for general unweighted graphs, and optimal algorithms for unweighted trees and the  $\{0, 1\}$ -EDS problem for weighted trees. To the

**Table 1** Approximation ratios for variants of the EDS problem (\* denotes a linear time algorithm)

	Unweighted EDS	Weighted EDS	Unweighted $b$ -EDS	Weighted $b$ -EDS
General graphs	2* [7]	2 [5, 13]	2* (Cor. 2)	8/3 [1]
Bipartite graphs	''	''	''	2 [1]
Trees	1* [11]	1* (Thm. 3)	1* (Thm. 2)	1 (Thm. 1)

best of our knowledge an exact linear time algorithm was not known for even the special case of weighted EDS on trees. Table 1 gives an overview of known results and new results from this paper.

*Organization* In Sect. 2 we expose a relationship between the maximum weighted matching problem and the unweighted  $b$ -EDS problem; we use this relationship to analyze an algorithm of Preis [16] and show that it is also a linear time 2-approximation for the unweighted  $b$ -EDS problem. This generalizes a known relationship between maximal matchings and (unweighted) edge dominating sets.

In Sect. 3 we show that the weighted  $b$ -EDS problem is solvable in polynomial time on trees. We also present exact linear-time algorithms which solve the unweighted  $b$ -EDS problem, and the weighted  $\{0, 1\}$ -EDS problem on trees. The latter is a primal-dual algorithm and the proof for the optimality of the returned solution shows that the algorithm can also generate an optimal dual solution. If the weights  $c(e)$  are integral for all  $e \in E$ , then the dual solution is integral as well and is a maximum size set of edges such that each edge  $e$  has at most  $c(e)$  edges adjacent to it. This problem is a common generalization of the maximum independent set problem and the maximum strong matching problem. An exact linear time algorithm for the latter on trees is known [3].

*Notation* We will use the following notation for a simple undirected graph  $G = (V, E)$ . The neighbors of a vertex  $v \in V$  are denoted by  $\delta(v) = \{u \in V : \exists uv \in E\}$ . The edges incident upon the vertex  $v \in V$  are denoted by  $N(v)$  and the set of edges adjacent with an edge  $e \in E$  plus the edge  $e$  itself is denoted by  $N(e)$ , i.e.  $N(uv) = N(u) \cup N(v)$  for any  $uv \in E$ .

## 2 The Unweighted $b$ -EDS Problem for General Graphs

For general graphs, the weighted EDS problem admits a 2.1-approximation based on a natural linear program relaxation of the problem, whose integrality gap is also 2.1 [2]. A corresponding linear relaxation for the weighted  $b$ -EDS problem yields an  $\frac{8}{3}$ -approximation for general graphs and a 2-approximation for bipartite graphs [14]. This relaxation can be strengthened to yield a 2-approximation for weighted EDS [5, 13]; however, the corresponding strengthening fails to deliver a 2-approximation for weighted  $b$ -EDS.

The unweighted EDS and  $b$ -EDS problems, however, can be approximated more easily due to their relation to matching problems. Harary's book [7] demonstrates that there always exists a minimum cardinality EDS which is also a maximal matching. Since any maximal matching is also an EDS, the minimum cardinality maximal matching and the unweighted EDS problems are equivalent. In contrast Fujito [4] showed that the minimum weighted maximal matching problem is much more difficult than weighted EDS. Any maximal matching in a graph has size at least one half times the size of a maximum matching. Therefore, finding any maximal matching, which can be easily done in linear time, yields a 2-approximation for the unweighted EDS problem.

An issue with extending the relationship described above to the minimum unweighted  $b$ -EDS problem is that a maximal matching is not necessarily a feasible

*b*-EDS. Using the resemblance of the maximum weight matching problem to the dual of a natural linear formulation for *b*-EDS, we exhibit a connection between weighted matchings and *b*-edge dominating sets that generalizes the relationship between maximal matchings and edge dominating sets. Given a matching *M* and a vector  $b \in \mathbb{N}^{|E|}$ , let  $b|_M \in \mathbb{N}^{|E|}$  denote the vector which for each component  $e \in M$  has value  $b_e$ , and has value 0 for all other components.

**Lemma 1** *For any matching  $M$  and any vector  $b \in \mathbb{N}^{|E|}$ ,  $\sum_{e \in M} b_e$  is at most twice the weight of a minimum size (counting multiplicities) *b*-EDS.*

*Proof* Consider the following pair of dual LP’s, LP 1 being the linear programming relaxation for the unweighted *b*-EDS problem and LP 2 being the relaxation for the weighted strong matching problem. We call *P* and *R* the

$$\begin{array}{ll} \text{LP 1: } \min \mathbb{1} \cdot x, \text{ subject to} & \text{LP 2: } \max b \cdot y, \text{ subject to} \\ x(N(e)) \geq b_e \quad \text{for all } e \in E & y(N(e)) \leq 1 \quad \text{for all } e \in E \\ x_e \geq 0 \quad \text{for all } e \in E & y_e \geq 0 \quad \text{for all } e \in E \end{array}$$

sets of feasible fractional solutions of LP 1 and LP 2, respectively. Let now *M* be a matching in the graph. By setting the dual variables to  $\bar{y}_e = 1/2$  for each  $e \in M$ , and to  $\bar{y}_e = 0$  for each  $e \in \setminus M$ , we obtain such a feasible solution  $\bar{y} \in R$  to LP 2, since each edge  $e \in E$  can be adjacent with at most two edges from *M*. Using duality we have

$$b(M)/2 = b \cdot \bar{y} \leq \max_{y \in R} b \cdot y = \min_{x \in P} \mathbb{1} \cdot x \leq OPT.$$

Hence the weight  $b(M)$  of the solution we return is at most  $2 \cdot OPT$ . □

**Corollary 1** *For any matching  $M$  and any vector  $b \in \mathbb{N}^{|E|}$ , if  $b|_M$  is a feasible *b*-EDS then it is a 2-approximate unweighted *b*-EDS.*

Corollary 1 motivates the following definition: we say a matching *M* is *b*-feasible if  $b|_M$  is a feasible *b*-EDS. Thus any matching algorithm that returns a *b*-feasible matching *M* is a 2-approximation for the unweighted *b*-EDS problem. Before presenting a linear time algorithm, we present a very simple  $O(|E| \cdot \log |V|)$  2-approximation for unweighted *b*-EDS.

**Proposition 1** *The greedy algorithm for the maximum weight matching problem (with weights  $b \in \mathbb{N}^{|E|}$ ) that repeatedly selects the edge of greatest weight that maintains a matching, always returns a matching that is also *b*-feasible.*

*Proof* The greedy algorithm also satisfies Lemma 2 below and thus the proof is the same as the proof of Lemma 3. □

It is not difficult to see that any matching of maximum weight with respect to *b* is *b*-feasible. Lemma 1 also implies that any feasible *b*-EDS has size at least  $\frac{1}{2}$  the cost of a maximum weight matching with respect to *b*, thus if a *b*-EDS algorithm always

returns a  $b$ -EDS that is a matching when copies of an edge are removed, then the algorithm is a  $\frac{1}{2}$ -approximation for the maximum weight matching problem.

Preis [16] gave a linear time algorithm, which, given a weighted graph, computes a maximal matching with weight at least one half times the weight of any matching. The algorithm incrementally adds edges to a matching  $M$ . A vertex  $u$  is called free (w.r.t. to  $M$ ), if  $u$  is not incident with any edge in  $M$ . We will use the following lemma, which gives a necessary condition for an edge to be added to the matching, to show that Preis's algorithm always generates a  $b$ -feasible matching.

**Lemma 2** ([16, Lemma 3]) *If an edge  $uv$  is added to  $M$  during the algorithm, then  $u$  and  $v$  are free and neither  $u$  nor  $v$  are adjacent to a free vertex with an edge of higher weight than the weight of the edge  $uv$ .*

**Lemma 3** *Preis's algorithm always generates a  $b$ -feasible matching.*

*Proof* First, note that each edge  $e \in M$  is covered  $b_e$  times by itself. Since  $M$  is a maximal matching, any edge  $e = uv$  in  $E \setminus M$  must be adjacent to some edge  $f = vw$  in  $M$ . If  $b_f \geq b_e$ , then  $e$  is covered. Otherwise, Lemma 2 says that  $u$  was not a free vertex at the time when  $f$  was added to  $M$ . But then there must be an edge  $tu \in M$ , which was added to  $M$  before  $f$ , i.e.  $t$ ,  $u$  and  $v$  were free vertices at the time  $tu$  was considered. Using Lemma 2 again, we must have  $b_{tu} \geq b_e$ .  $\square$

**Corollary 2** *The unweighted  $b$ -EDS problem on general graphs can be 2-approximated in linear time.*

### 3 The $b$ -EDS Problem for Trees

#### 3.1 The General Case

Many problems which are hard to solve optimally or even approximate for general graphs become considerably easier when restricted to a small family of graphs. The same is true for the weighted  $b$ -EDS problem when we restrict the possible inputs to trees.

In this section we will show that the weighted  $b$ -EDS problem on trees can be solved optimally as a linear program. A square matrix  $A$  is called *totally unimodular* if every square sub-matrix of  $A$  has determinant  $+1$ ,  $-1$  or  $0$ . Totally unimodular matrices play an important role in linear programming due to the following lemma.

**Lemma 4** ([8, 17]) *If  $A \in \mathbb{Z}^{m \times n}$  is a totally unimodular matrix and  $b \in \mathbb{Z}^m$ , then every extreme point of the polyhedron  $\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\} \neq \emptyset$  has integer coordinates.*

Ghouila-Houri [6] gave the following sufficient condition for a matrix to be totally unimodular.

**Lemma 5** *An  $m \times n$  matrix  $A$  with entries in  $\{1, 0, -1\}$  is totally unimodular if for every  $J \subseteq \{1, \dots, n\}$  there exist a partition  $J = J_1 \cup J_2$  such that for any  $1 \leq j \leq m$  it holds that  $|\sum_{i \in J_1} a_{ij} - \sum_{i \in J_2} a_{ij}| \leq 1$ .*

We now prove that the constraint matrix for the EDS problem for any tree is totally unimodular. To the best of our knowledge this is the first proof of this fact in the literature.

**Lemma 6** *Let  $T = (V, E)$  be a tree on  $n$  vertices. Let  $A = (a_{ij})$  be the edge-edge adjacency matrix of  $T$  with 1's on the diagonal, i.e.  $a_{ij} = 1$  for all  $ij \in E$ ,  $a_{ij} = 0$  for all  $ij \notin E$  and  $a_{ii} = 1$  for all  $1 \leq i \leq n - 1$ ,  $1 \leq j \leq n - 1$  and  $i \neq j$ . Then  $A$  is totally unimodular.*

*Proof* According to Lemma 5 it is enough to show that we can partition any  $E' \subseteq E$  into two sets of edges  $E_1$  and  $E_2$ , so that for any  $e \in E$  we have  $|N(e) \cap E_1| - |N(e) \cap E_2| \in \{-1, 0, 1\}$ . This is equivalent to saying that for any  $E' \subseteq E$  there is a labeling  $\pi : E \rightarrow \{1, 0, -1\}$  such that  $\pi(e) = 0$  if and only if  $e \in E \setminus E'$  and that  $\Pi(e) := |\sum_{f \in N(e)} \pi(f)| \leq 1$  for any  $e \in E$ .

We will prove by induction on  $|V|$  that such a labeling exists for any  $E' \subseteq E$  and that it also satisfies  $|\Pi(v)| \leq 1$  where  $\Pi(v) := \sum_{f \in N(v)} \pi(f)$  for any  $v \in V$ . Note that if such a labeling  $\pi$  exists, then  $-\pi$  clearly also satisfies the condition.

The base case  $|V| = 1$  is trivial. Let now  $|V| \geq 2$  and let  $v_0 \in V$  be an arbitrary vertex of  $T$ . Let  $T_i = (V_i, E_i)$  ( $1 \leq i \leq k$ ) be the connected components of  $T - \{v_0\}$ , which are trees as well. Let  $v_i$  be the neighbor of  $v_0$  in  $T_i$  and  $e_i = v_0v_i$ . For every  $1 \leq i \leq k$  let  $\pi_i$  be a labeling of  $E_i$  with  $\pi_i(e) = 0$  if and only if  $e \in E_i \setminus E'$ , which exists by the inductive hypothesis. They also satisfy  $|\Pi_i(v_i)| \leq 1$  for every  $1 \leq i \leq k$ . If we set  $\pi(e_i) = 0$  for  $1 \leq i \leq k$  and  $\pi(e) = \pi_i(e)$  whenever  $e \in E_i$ , we obtain a labeling  $\pi : E \rightarrow \{1, 0, -1\}$  such that  $\Pi(e_i) = |\Pi_i(v_i)| \leq 1$  and  $\Pi(v_0) = 0$ , i.e.  $\pi$  satisfies the claim. However, the edges incident with  $v_0$  which are in  $E'$  are falsely labeled 0. We will show how to label these edges with 1 or  $-1$  and maintain the desired properties.

We can assume w.l.o.g. that for some  $0 \leq s \leq k$  we have that  $\{e_i : 1 \leq i \leq s\} \subseteq E'$  and  $\{e_i : s + 1 \leq i \leq k\} \subseteq E \setminus E'$ . Further assume that  $|\Pi(v_i)| = 1$  for  $1 \leq i \leq s_0$  and that  $|\Pi(v_i)| = 0$  for  $s_0 < i \leq s$  for some  $0 \leq s_0 \leq s$ .

We will switch the labellings  $\pi_i$  for  $1 \leq i \leq s_0$  if necessary to obtain  $\Pi(v_i) = 1$  for  $1 \leq i \leq \lfloor s_0/2 \rfloor$  and  $\Pi(v_i) = -1$  for  $\lfloor s_0/2 \rfloor < i \leq s_0$ . Then we define  $\pi(e_i) = -1$  for  $1 \leq i \leq \lfloor s_0/2 \rfloor$  and  $\pi(e_i) = 1$  for  $\lfloor s_0/2 \rfloor < i \leq s_0$ . The edges  $e_{s_0+1}, \dots, e_s$  will be labeled with 1 and  $-1$  so that we have  $\Pi(v_0) \in \{0, 1\}$ . The numbers of edges labeled 1 and  $-1$ , respectively, will depend on the parity of  $s$  and  $s_0$  and differ by at most 1. If  $\Pi(v_0) = 1$ , then we also switch the labellings  $\pi_i$  of those trees  $T_i$  with  $i > s$  for which  $\Pi(v_i) = 1$  to  $-\pi_i$  to ensure that  $\Pi(e_i) \leq 1$  for those indices  $i$ .

It is easy to check that the conditions on  $\pi$  remain true for all edges and vertices of  $T$ . □

Using Lemma 4 and Lemma 6 we immediately have

**Theorem 1** *The  $b$ -EDS problem on weighted trees can be solved optimally in strongly polynomial time.*

The algorithm to solve the  $b$ -EDS problem on trees relies on solving a linear program. However, we would prefer a combinatorial algorithm, ideally running in linear time. This is indeed possible if we restrict the trees to have either uniform weights (Sect. 3.2) or if we restrict ourselves to the  $\{0, 1\}$ -EDS problem on weighted trees.

### 3.2 The Unweighted $b$ -EDS Problem for Trees

A linear time algorithm for the unweighted EDS problem on trees was first given by Mitchell and Hedetniemi [11] and later simplified by Yannakakis and Gavril [20]. The unweighted  $b$ -EDS problem on trees can also be solved by an easy greedy algorithm in linear time. Call an edge  $e$  of a tree a *leaf edge* if it is incident with a leaf.

For any tree  $T$  there will always be an optimal solution to the  $b$ -EDS problem which does not use any leaf edges (unless  $T$  is a star), since any edge adjacent with a leaf edge covers at least those edges covered by the leaf edge.

Therefore we can recursively solve the problem by first finding a vertex  $v$  which is incident with exactly one non-leaf edge  $e$ , then setting the multiplicity of  $e$  to the maximum  $b$ -value of the leaf edges incident with  $v$  and finally removing those leaf edges and updating the  $b$ -values of  $e$  and those edges adjacent with  $e$  in the remaining tree.

Any optimal solution to the  $b$ -EDS problem on that updated tree plus the multiplicity of the edge  $e$  as determined before will give an optimal solution to the original instance. A formal proof of this fact is straightforward and we omit it in here.

**Theorem 2** *The unweighted  $b$ -EDS problem on trees can be solved optimally in linear time.*

### 3.3 The Weighted $\{0, 1\}$ -EDS Problem for Trees

To the best of our knowledge no linear time algorithm for the weighted EDS problem on trees has appeared in the literature. Algorithm 1 is a linear time primal-dual algorithm, which solves the weighted  $\{0, 1\}$ -EDS problem on trees (indeed a generalization of the weighted EDS problem) optimally in linear time.

For a rooted tree and a vertex  $v$ , we denote by  $ch(v)$  the children of  $v$ , by  $gch(v)$  the grandchildren of  $v$ , by  $E_{ch}(v)$  the edges between  $v$  and its children, and by  $E_{gch}(v)$  all edges between children and grandchildren of  $v$ . Moreover,  $gp(v)$  denotes the grandparent of  $v$ .

Our algorithm uses a primal-dual approach. We consider the tree to be rooted at a root vertex  $v_0$ . Starting with a leaf vertex  $v$ , we consider the edges incident with the parent vertex of  $v$  and their respective  $b$  and  $c$ -values. Depending on these values, we modify the  $b$  and  $c$  values of these edges and reduce the instance by removing edges incident with the parent of  $v$ . The changes made will be used to set the dual solution.

**Algorithm 1** EDS( $T, b, c$ ) for  $\{0, 1\}$ -EDS

**Input:** An instance  $(T = (V, E), b, c)$  of the  $\{0, 1\}$ -EDS problem, where  $T$  is a tree rooted at some  $v_0 \in V$ .

**Output:** A minimum cost  $\{0, 1\}$ -EDS of  $T$ .

```

1: if  $T$  is a star rooted at  $v_0$  then
2:   return  $\emptyset$  if  $b(e) = 0$  for all  $e \in E$  else return  $\operatorname{argmin}\{c(e) : e \in E\}$ 
3: else
4:   Let  $v$  be a leaf of maximum depth and  $w = gp(v)$ .
5:   if  $b(e) = 0$  for all  $e \in E_{gch}(w)$  then
6:      $S = \bigcup_{s \in ch(w): b(ws)=1} \operatorname{argmin}\{c(e) : e \in E_{ch}(s)\}; \alpha = c(S)$ 
7:      $f = \operatorname{argmin}\{c(e) : e \in N(w)\}; \varepsilon = c(f); \beta = \min\{\alpha, \varepsilon\}$ 
8:      $c'(e) = c(e) - \beta$  for all  $e \in N(w)$ 
9:     if  $\alpha < \varepsilon$  then
10:       $b'(e) = 0$  for all  $e \in E_{ch}(w); S' = S$ 
11:     else
12:       $b'(e) = 0$  for all  $e \in N(w); S' = \{f\}$ 
13:      $F = \text{EDS}(T - gch(w), b', c')$ 
14:     if  $F \cap N(w) = \emptyset$  then
15:       return  $F \cup S'$  else return  $F$ 
16:   else
17:     Let  $s \in ch(w)$  and  $t \in gch(w)$  be such that  $b(st) = 1$ .
18:     if  $c(ws) < \min\{c(e) : e \in E_{ch}(s)\}$  then
19:        $b'(e) = 0$  for all  $e \in N(w); c'(ws) = 0$ 
20:       return  $\text{EDS}(T - ch(s), b', c') \cup \{ws\}$ 
21:      $f = \operatorname{argmin}\{c(e) : e \in N(s)\}$ 
22:      $b'(ws) = 0; c'(ws) = c(ws) - c(f); F = \text{EDS}(T - ch(s), b', c')$ 
23:     if  $ws \in F$  then
24:       return  $F$  else return  $F \cup \{f\}$ 

```

After solving the problem on the reduced and modified instance, it is possible to construct an optimal solution for the original instance using these dual values. The basic idea for this is similar to the argument used for the unweighted case in Sect. 3.2 above. However, having non-uniform weights may result in edges incident upon a leaf being part of the optimal solution. Therefore, there will be more cases to be considered, and, in particular, we will have to make sure that the algorithm still runs in linear time. The details are shown in the proof of the following theorem.

**Theorem 3** Algorithm 1 solves the weighted  $\{0, 1\}$ -EDS problem on trees optimally.

*Proof* Let  $F^*$  be the solution returned by the algorithm. Due to the recursive nature of the algorithm one can verify (for all cases that can occur) that  $F^*$  is indeed a feasible  $\{0, 1\}$ -EDS of the instance  $(T, b, c)$  by using the fact that the recursively obtained solution  $F$  is a feasible solution to the reduced instance.



To prove the optimality of  $F^*$  we will show that there exists a dual solution  $y$  with dual variables  $y(e)$  for all  $e$  with  $b(e) = 1$ , such that  $\mathbb{1} \cdot y = c(F^*)$ , which proves the optimality of  $F^*$ . Remember that the dual constraints are  $y(N(e)) \leq c(e)$  for all  $e \in E$ .

We will prove this claim by induction on the size of  $T$ . The claim is trivially true in the base case, i.e. when  $T$  is a star rooted at  $v_0$ .

Let us now assume  $T$  is not a star rooted at  $v_0$ . All variables used in this proof are as they are defined in the algorithm.

We first consider the case when  $b(e) = 0$  for all  $e \in E_{gch}(w)$ . Let  $y'$  be a feasible solution to the dual of the instance  $\mathbf{EDS}(T, b', c')$ , which has the property that  $c'(F) = \mathbb{1} \cdot y'$ . Such a  $y'$  exists by the inductive hypothesis. In all cases (whether  $\alpha < \varepsilon$  or not and whether  $F \cap N(w) = \emptyset$  or not) we will have that  $c(F^*) = c'(F) + \beta$ , i.e. to prove the claim we have to be able to raise the  $y'$ -value of some edges by a total of  $\beta$  while still keeping  $y'$  a feasible solution to the dual of the instance  $\mathbf{EDS}(T, b, c)$ . There are several cases to consider. If  $F \cap N(w) \neq \emptyset$ , then we can raise the dual variable of an arbitrary edge incident upon  $w$  by  $\beta$ . The new dual solution is feasible for the dual of  $\mathbf{EDS}(T, b, c)$ , since  $c'(e) = c(e) - \beta$  for all  $e \in N(w)$  and because  $y'$  was feasible for the dual of  $\mathbf{EDS}(T, b', c')$ .

In the other case ( $F^* = F \cup S'$  and  $\beta = \alpha$ ), we raise the  $y'$ -value of each edge  $ws$  with  $s \in ch(w)$  and  $b(ws) = 1$  by  $\min\{c(e) : e \in E_{ch}(s)\}$ . In the way  $\beta$  was defined, this raises the  $y'$ -values of edges incident upon  $w$  by a total of  $\beta$ . Now the dual constraints are satisfied for all edges in  $E_{gch}(w)$ . Further, since  $c'(e) = c(e) - \beta$  for all  $e \in N(w)$  and since  $y'$  was feasible for the dual of  $\mathbf{EDS}(T, b', c')$ , the new (raised) dual variables are also feasible for the dual of  $\mathbf{EDS}(T, b, c)$ .

We finally have to consider the case when  $b(st) = 1$  for some  $s \in ch(w)$  and  $t \in gch(w)$  (lines 17–25 in the algorithm). Let  $y'$  be a feasible solution to the dual of the instance  $\mathbf{EDS}(T - ch(s), b', c')$ , which has the property that  $c'(F) = \mathbb{1} \cdot y'$ .

If the condition in line 18 is satisfied, we can set  $y'(st) = c(ws)$  to obtain a feasible solution to the dual of  $\mathbf{EDS}(T, b, c)$ . We raised the value of  $\mathbb{1} \cdot y'$  by  $c(ws)$  and also the value of the solution by  $c(ws)$  by adding the edge  $ws$  to the solution (line 20). On the other hand, if the latter condition is not satisfied, then we can set  $y'(st) = c(f)$  and the same analysis applies.  $\square$

The algorithm can be implemented to run in linear time using a BFS ordering of the vertices that lets us easily find a leaf of maximum depth in the rooted tree.

The weighted  $\{0, 1\}$ -EDS problem generalizes some well-known other domination problems. Therefore Algorithm 1 also provides linear-time algorithms for these problems on trees. We start with the weighted vertex cover problem.

**Lemma 7** *The weighted vertex cover problem for trees can be solved optimally in linear time by solving a weighted  $\{0, 1\}$ -EDS instance on a tree in linear time.*

*Proof* Let  $T = (V, E)$ ,  $c_v \in \mathbb{R}^+$  for all  $v \in V$  be an instance of the weighted vertex cover problem for trees. We build a tree  $T' = (V \cup V', E \cup E')$  with  $V' = \{v' : v \in V\}$  and  $E' = \{vv' : v \in V\}$ , i.e. we add an extra edge incident with each vertex of  $T$ . We set  $b_e = 1$  for every  $e \in E$  and  $b_e = 0$  for every  $e \in E'$ . Furthermore, we set  $c'_{vv'} = c_v$

for all  $v \in V$  and  $c'_e = \infty$  for all  $e \in E$ . Then any  $\{0, 1\}$ -EDS of  $T'$  of finite weight corresponds to a vertex cover of  $T$  having the same weight, and vice versa (an edge  $vv' \in E'$  is in the  $\{0, 1\}$ -EDS if and only if  $v$  is in the vertex cover).  $\square$

Moreover, our algorithm can also be used to solve the weighted edge-cover problem for trees optimally in linear time.

**Lemma 8** *The weighted edge cover problem for trees can be solved optimally in linear time by solving a weighted  $\{0, 1\}$ -EDS instance on a tree in linear time.*

*Proof* Let  $T = (V, E)$ ,  $c_e \in \mathbb{R}^+$  for all  $e \in E$  be an instance of the weighted edge cover problem for trees. We build a tree  $T' = (V \cup V', E \cup E')$  with  $V' = \{v' : v \in V\}$  and  $E' = \{vv' : v \in V\}$ , i.e. we add an extra edge incident with each vertex of  $T$ . We set  $b_e = 0$  for every  $e \in E$  and  $b_e = 1$  for every  $e \in E'$ . Furthermore, we set  $c'_{vv'} = \infty$  for all  $v \in V$  and  $c'_e = c_e$  for all  $e \in E$ . Then any  $\{0, 1\}$ -EDS of  $T'$  of finite weight corresponds to an edge cover of  $T$  having the same weight, and vice versa (an edge  $e \in E'$  is in the  $\{0, 1\}$ -EDS if and only if  $e$  is in the edge cover).  $\square$

## 4 Conclusion

In this paper we considered the  $b$ -edge dominating set problem. We provided a linear-time 2-approximation algorithm for this problem in general unweighted graphs, and several optimal algorithms for some cases of this problem on trees.

As Theorem 1 shows, the weighted  $b$ -EDS problem on trees for a general function  $b$  can be solved using a linear programming formulation of the problem. However, we do conjecture the following:

**Conjecture 1** *The weighted  $b$ -EDS problem on trees can be solved optimally in linear time.*

We also showed that the weighted  $b$ -edge dominating set problem generalizes the weighted vertex cover problem and the weighted edge-cover problem. In addition we mention, that the maximum weight matching problem can be reduced to the weighted edge cover problem in linear time [15] and therefore Algorithm 1 can also be used to solve the maximum weight matching problem for trees in linear time.

## References

1. Berger, A., Fukunaga, T., Nagamochi, H., Parekh, O.: Capacitated  $b$ -edge dominating set and related problems. Manuscript (2005)
2. Carr, R., Fujito, T., Konjevod, G., Parekh, O.: A 2 1/10-approximation algorithm for a generalization of the weighted edge-dominating set problem. *J. Comb. Optim.* **5**, 317–326 (2001)
3. Fricke, G., Laskar, R.: Strong matchings on trees. In: Proceedings of the Twenty-third Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Boca Raton, FL, 1992, vol. 89, pp. 239–243 (1992)

4. Fujito, T.: On approximability of the independent/connected edge dominating set problems. *Inf. Process. Lett.* **79**(6), 261–266 (2001)
5. Fujito, T., Nagamochi, H.: A 2-approximation algorithm for the minimum weight edge dominating set problem. *Discrete Appl. Math.* **118**(3), 199–207 (2002)
6. Ghouila-Houri, A.: Caractérisation des matrices totalement unimodulaires. *C. R. Acad. Sci. Paris* **254**, 1192–1194 (1962)
7. Harary, F.: *Graph Theory*. Addison-Wesley, Reading (1969)
8. Hoffman, A.J., Kruskal, J.B.: Integral boundary points of convex polyhedra. In: *Linear Inequalities and Related Systems*. *Annals of Mathematics Studies*, vol. 38, pp. 223–246. Princeton University Press, Princeton (1956)
9. Horton, J.D., Kilakos, K.: Minimum edge dominating sets. *SIAM J. Discrete Math.* **6**(3), 375–387 (1993)
10. Hwang, S.F., Chang, G.J.: The edge domination problem. *Discuss. Math. Graph Theory* **15**(1), 51–57 (1995)
11. Mitchell, S.L., Hedetniemi, S.T.: Edge domination in trees. In: *Proc. of the 8th Southeastern Conference on Combinatorics, Graph Theory and Computing*, Louisiana State Univ., Baton Rouge, LA, 1977, pp. 489–509 (1977)
12. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.* **43**(3), 425–440 (1991)
13. Parekh, O.: Edge dominating and hypomatchable sets. In: *Proceedings of the 13th Annual ACM-SIAM Symposium On Discrete Mathematics (SODA-02)*, pp. 287–291. ACM Press, New York (2002)
14. Parekh, O.: Polyhedral techniques for graphic covering problems. Ph.D. thesis, Carnegie Mellon University (2002)
15. Parekh, O., Razouk, N.: A generalization of Gallai’s theorem. Manuscript (2005)
16. Preis, R.: Linear time  $\frac{1}{2}$ -approximation algorithm for maximum weighted matching in general graphs. In: *STACS 99 (Trier)*. *Lect. Notes in Comp. Sci.*, vol. 1563, pp. 259–269. Springer, Berlin (1999)
17. Schrijver, A.: *Combinatorial Optimization. Polyhedra and Efficiency*. Algorithms and Combinatorics, vol. 24. Springer, Berlin (2003)
18. Srinivasan, A., Madhukar, K., Nagavamsi, P., Pandu Rangan, C., Chang, M.-S.: Edge domination on bipartite permutation graphs and cotriangulated graphs. *Inf. Process. Lett.* **56**(3), 165–171 (1995)
19. Vazirani, V.: *Approximation Algorithms*. Springer, New York (2001)
20. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. *SIAM J. Appl. Math.* **38**(3), 364–372 (1980)