



Improved Budgeted Connected Domination and Budgeted Edge-Vertex Domination

Ioannis Lamprou, Ioannis Sigalas, and Vassilis Zissimopoulos^(✉)

Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Zografou, Greece
{ilamprou, sigalasi, vassilis}@di.uoa.gr

Abstract. We consider the *Budgeted* version of the classical *Connected Dominating Set* problem (BCDS). Given a graph G and a budget k , we seek a connected subset of at most k vertices maximizing the number of dominated vertices in G . We improve over the previous $(1 - 1/e)/13$ approximation in [Khuller, Purohit, and Sarpatwar, *SODA 2014*] by introducing a new method for performing tree decompositions in the analysis of the last part of the algorithm. This new approach provides a $(1 - 1/e)/12$ approximation guarantee. By generalizing the analysis of the first part of the algorithm, we are able to modify it appropriately and obtain a further improvement to $(1 - e^{-7/8})/11$. On the other hand, we prove a $(1 - 1/e + \epsilon)$ inapproximability bound, for any $\epsilon > 0$.

We also examine the *edge-vertex domination* variant, where an edge dominates its endpoints and all vertices neighboring them. In *Budgeted Edge-Vertex Domination* (BEVD), we are given a graph G , and a budget k , and we seek a, not necessarily connected, subset of k edges such that the number of dominated vertices in G is maximized. We prove there exists a $(1 - 1/e)$ -approximation algorithm. Also, for any $\epsilon > 0$, we present a $(1 - 1/e + \epsilon)$ -inapproximability result by a gap-preserving reduction from the *maximum coverage* problem. Finally, we examine the “dual” *Partial Edge-Vertex Domination* (PEVD) problem, where a graph G and a quota n' are given. The goal is to select a minimum-size set of edges to dominate at least n' vertices in G . In this case, we present a $H(n')$ -approximation algorithm by a reduction to the *partial cover* problem.

Keywords: Approximation · Budget · Partial · Connected domination · Edge-vertex domination

1 Introduction

The problem of vertices dominating vertices in a graph is very common and has been extensively studied in graph theory and combinatorial optimization

This work was partially supported by the Special Account for Research Grants (ELKE) of the National and Kapodistrian University of Athens (NKUA). A full version of the paper can be found at <https://arxiv.org/abs/1907.06576>.

literature. In the classical definition, a dominating set is a subset of vertices such that each vertex is either a member of the subset or adjacent to a member of the subset. Intuitively, a dominating set provides a skeleton for the placement of resources, such that any network node is within immediate reach to them.

However, as it is often the case, there are constraints on the amount of resources available for placement, e.g., due to financial or other management reasons. That is, we are limited to a budget of k resources to be placed on network nodes. The optimization goal is to place the available resources suitably, such that the number of network nodes they dominate is maximized. This problem is known in literature as the *Budgeted Dominating Set* problem.

Budgeted domination has applications especially in ad-hoc wireless (sensor) networks. In this setting, a set of network nodes needs to be identified as the virtual backbone of the network, that is, the structure responsible for routing and packet forwarding. To achieve these tasks, nodes in the backbone must be able to communicate with each other, i.e., form a *connected* set of vertices in the graph capturing the topology of their communication ranges. The resulting optimization problem is the *Budgeted Connected Dominating Set* (BCDS) problem. In this paper, we study BCDS and present an improved guarantee over the previous state of the art [12].

Besides BCDS, we examine other problems where graph edges are selected as dominators. The concept of edges dominating adjacent edges has been well-considered in literature; e.g., see [8, 27] for some preliminary results. An example application is in network tomography where probes need to be placed to monitor the health of network links [14].

In this paper, we consider cases where resources must be positioned on the links of a network to dominate network nodes. For instance, consider a power system where a limited number of static var compensators need to be placed on transmission lines' midpoints to locate faults affecting a big proportion of buses [10]. Another example is to identify a limited-size set of friendships, modeled as graph edges, having a big impact in terms of neighborhood in a social network.

More formally, the notion in consideration is *edge-vertex domination*, where an edge dominates its endpoints and any vertices adjacent to its endpoints. We examine the (in)approximability of *Budgeted Edge-Vertex Domination* (BEVD), where we seek a, not necessarily connected, set of k (budget) edges dominating as many vertices as possible. If the edge set is required to be connected, we show that the problem essentially matches BCDS. Finally, we consider the related *Partial Edge-Vertex Domination* (PEVD) problem: a quota of vertices needs to be dominated by utilizing the minimum number of edges possible.

1.1 Related Work

Finding a minimum-size connected set of vertices dominating the whole graph is a classical NP-hard problem. In [7], Guha and Khuller proposed a $\ln \Delta + 3$ approximation algorithm, which is (up to constant factors) the best possible, since the problem is hard to approximate within a factor of $(1 - \epsilon) \log n$ [5]. For

a bigger picture of the research landscape, in [4], many connected domination results for special graph classes and other applications are surveyed.

In [21], vertex-vertex and edge-edge budgeted domination are considered. For vertex-vertex, matching upper and lower bounds of $(1 - 1/e)$ are given, whereas, for edge-edge, a $(1 - 1/e)$ approximation and a $1303/1304 + \epsilon$ hardness are proved.

In the connected case, budgeted and partial versions of domination have their origins in wireless sensor networking [19, 26], where a network backbone with good qualities needs to be determined, which must either be limited in resources and/or cover a big-enough proportion of the network. The first, and thus far state of the art, results for the budgeted and partial cases in general graphs appear in [12], where a $(1 - 1/e)/13$ -approximation, respectively an $O(\ln \Delta)$ -approximation, is proved for the budgeted, respectively partial, case. Other works have followed in particular settings. For example, in [20], a constant factor approximation algorithm for partial connected domination on a superset of unit disk graphs, namely growth-bounded graphs, is proposed. Their result translates to a 27-approximation guarantee on unit disk graphs.

Regarding edge-vertex domination, the graph-theoretic notion was introduced in [22], together with the complementary case of vertex-edge domination, where a vertex dominates all edges incident to it or to a neighbor of it. Some complexity and algorithmic results about the minimal size of an edge-vertex, respectively vertex-edge, dominating set appear in [18]. More recently, some vertex-edge domination open questions posed in [18] were answered in [2]. In [25], an improved bound on the edge-vertex domination number of trees was proved. Except for the vertex-edge and edge-vertex variants, a *mixed* domination variant has been introduced [23], where a minimal subset of both vertices and edges need to be selected so that each vertex/edge of the graph is incident/adjacent to a vertex/edge in the subset. Recent example works in this topic study the problem in special graph classes like trees, cacti, and split graphs [17, 28].

1.2 Our Results

In Sect. 2, we present preliminary notions and formally define the problems.

In Sect. 3, we examine the Budgeted Connected Dominating Set (BCDS) problem, see Definition 1, where a connected subset of budget vertices needs to dominate as many vertices as possible. By introducing a new tree decomposition technique in Subsect. 3.2, we prove a $(1 - 1/e)/12 \simeq 0.05267$ approximation, in Theorem 2, which improves over the previous best known $(1 - 1/e)/13$ guarantee [12]. (We note the same guarantee has recently been achieved independently in [13].) We further improve the ratio to $(1 - e^{-7/8})/11 \simeq 0.05301$ (Theorem 3) by generalizing the first part of the analysis in [12] and then modifying the proposed algorithm accordingly in Subsect. 3.3. On the negative side, for any $\epsilon > 0$, we show a first $(1 - 1/e + \epsilon)$ inapproximability bound; see Theorem 5.

In Sect. 4, we consider edge-vertex domination, where a, not necessarily connected, subset of edges dominates adjacent vertices. If the set of edges is also required to be connected, then the problems essentially reduce to the standard vertex-vertex budgeted/partial dominating set problems; see Proposition 2. In

Subsect. 4.1, we prove there is a $(1-1/e)$ -approximation algorithm (Theorem 7). This is the best possible since we prove an $(1-1/e+\epsilon)$ inapproximability lower bound, for any $\epsilon > 0$, see Theorem 8. In Subsect. 4.2, we consider the problem of Partial Edge-Vertex Domination. In Theorem 10, we prove that, in the general case, there exists an $H(n')$ -approximation, where $H(\cdot)$ is the Harmonic number and n' is the number of vertices requested to be dominated. To do so, we employ a reduction to a partial version of the classical *Set Cover* problem.

Finally, in Sect. 5, we give some concluding remarks.

2 Preliminaries

A graph G is denoted as a pair $(V(G), E(G))$ (or simply (V, E)) of the vertices and edges of G . The graphs considered are simple (neither loops nor multi-edges are allowed), connected and undirected. Besides the aforementioned, no assumptions are made on the topology of the input graphs.

Two vertices $u, v \in V$ connected by an edge, denoted (u, v) or equivalently (v, u) , are called *adjacent* or *neighboring*. The *open neighborhood* of a vertex $v \in V$ is defined as $N(v) = \{u \in V : (v, u) \in E\}$, while the *closed neighborhood* is defined as $N[v] = \{v\} \cup N(v)$. For a subset of vertices $S \subseteq V(G)$, we expand the above definitions to $N(S) = \bigcup_{v \in S} N(v) \setminus S$ and $N[S] = N(S) \cup S$.

The degree of a vertex $v \in V$ is defined as $d(v) = |N(v)|$. The minimum, resp. maximum, degree of G is denoted by $\delta = \min_{v \in V} d(v)$, resp. $\Delta = \max_{v \in V} d(v)$.

Let us now consider the neighborhood of edges in terms of vertices. Given an edge $e = (v, u) \in E$, let $I(e) = \{v, u\}$ stand for the set containing its two incident vertices. We define the *neighborhood of an edge* e as $N[e] = \bigcup_{v \in I(e)} N[v]$. For a set of edges $E' \subseteq E$, we define $V(E') = \{v \in V \mid \exists e \in E' \text{ such that } v \in I(e)\}$. Then, we define the *edge-set neighborhood* as $N[E'] = N[V(E')]$. Here, we focus on a closed neighborhood definition, since it captures the number of vertices incident or adjacent to a set of edges in the standard edge-vertex domination paradigm (Definition 8 in [18]; originally introduced in [22]). That is, we say that a set of edges E' *dominates* $N[E']$.

Let us now proceed to formally define the problems studied in this paper.

Definition 1 (BUDGETED CONNECTED DOMINATING SET).

Given a graph $G = (V, E)$ and an integer k , select a subset $S \subseteq V$, where $|S| \leq k$, such that the subgraph induced by S is connected and $|N[S]|$ is maximized.

Definition 2 (BUDGETED EDGE-VERTEX DOMINATION).

Given a graph $G = (V, E)$ and an integer k , select a subset $E' \subseteq E$, where $|E'| \leq k$, such that $|N[E']|$ is maximized.

Definition 3 (PARTIAL EDGE-VERTEX DOMINATION).

Given a graph $G = (V, E)$ and an integer n' , select a subset $E' \subseteq E$ of minimum size such that it holds $|N[E']| \geq n'$.

3 Budgeted Connected Dominating Set

In this section, we consider the Budgeted Connected Dominating Set (BCDS) problem given in Definition 1. We initially present a summary of key aspects of the state of the art algorithm [12], which achieves a $(1 - 1/e)/13$ approximation factor. We then show how the analysis can be improved to achieve a $(1 - 1/e)/12$ guarantee via an alternative tree decomposition scheme; see Theorem 2. Then, we generalize the analysis of the greedy procedure in order to modify a call within the state of the art algorithm. This modification allows us to increase the approximation factor even further to $(1 - e^{-7/8})/11$; see Corollary 1. On the other hand, we conclude this section with a $(1 - 1/e + \epsilon)$, for any $\epsilon > 0$, inapproximability result; see Theorem 5.

3.1 Previous Approach

Khuller et al., see Algorithm 2 (Algorithm 5.1 in [12]), design the first constant factor approximation algorithm for BCDS with an approximation guarantee of $(1 - 1/e)/13$. Their approach comprises three method calls: (i) a call to an algorithm returning a greedy dominating set D and its corresponding profit function p ; see Algorithm 1 (GDS), (ii) a call to a 2-approximation algorithm, which follows from [6,9], for the *Quota Steiner Tree* (QST) problem defined below, and (iii) a call to a dynamic programming scheme $Best_k(\cdot)$ to determine the maximum-profit subtree of size at most k within a bigger-size tree.

Algorithm 1: Greedy Dominating Set (GDS) [12]

```

Input  : A graph  $G = (V(G), E(G))$ 
Output: A dominating set  $D \subseteq V(G)$  and a profit function
            $p : V(G) \rightarrow \mathbb{N} \cup \{0\}$ 
1  $D \leftarrow \emptyset$ 
2  $U \leftarrow V(G)$ 
3 foreach  $v \in V(G)$  do
4   |  $p(v) \leftarrow 0$ 
5 end
6 while  $U \neq \emptyset$  do
7   |  $w \leftarrow \arg \max_{v \in V(G) \setminus D} |N_U(v)|$            /*  $N_U(v) = N[v] \cap U$  */
8   |  $p(w) \leftarrow |N_U(w)|$ 
9   |  $U \leftarrow U \setminus N_U(w)$ 
10  |  $D \leftarrow D \cup \{w\}$ 
11 end
12 return  $(D, p)$ 

```

Definition 4 (QUOTA STEINER TREE). *Given a graph G , a vertex profit function $p : V(G) \rightarrow \mathbb{N} \cup \{0\}$, an edge cost function $c : E(G) \rightarrow \mathbb{N} \cup \{0\}$ and a quota $q \in \mathbb{N}$, find a subtree T that minimizes $\sum_{e \in E(T)} c(e)$ subject to the condition $\sum_{v \in V(T)} p(v) \geq q$.*

Algorithm 2: Greedy Profit Labeling Algorithm for BCDS [12]

Input : A graph $G = (V(G), E(G))$ and $k \in \mathbb{N}$
Output: A tree \tilde{T} on at most k vertices

- 1 $(D, p) \leftarrow GDS(G)$
- 2 $T \leftarrow QST(G, (1 - 1/e)OPT, p)$
- 3 $\tilde{T} \leftarrow Best_k(T, p)$
- 4 **return** \tilde{T}

Theorem 1 (Follows from results in [6, 9]). *There is a 2-approximation algorithm for QUOTA STEINER TREE.*

In their analysis, Khuller et al. [12] demonstrate that there exists a set $D' \subseteq D$ of size k which dominates at least $(1 - 1/e)OPT$ vertices, where OPT is the optimal number of dominated vertices achieved with a budget of k . Furthermore, D' can be connected by adding at most another $2k$ Steiner vertices, so giving a total of $3k$ vertices. Then, it suffices to call the 2-approximation algorithm for QST, see line 2 in Algorithm 2, with profit function p (returned by algorithm GDS at line 1), all edge costs equal to 1 and quota equal to $(1 - 1/e)OPT$. The value OPT can be guessed via a binary search between k and n . Overall, the returned tree has size at most $6k$ vertices and dominates at least $(1 - 1/e)OPT$ vertices: a $(6, 1 - 1/e)$ bicriteria approximation is attained (Lemma 5.2 [12]).

As a final step ($Best_k(\cdot)$ at line 3), a dynamic programming approach is used to identify the best-profit subtree with at most k vertices, such that the budget requirement is satisfied; see paragraph 5.2.2 in [12] for the relevant recurrences. To obtain a true approximation guarantee for the final solution, the following tree decomposition lemma is used recursively to prove that, for a sufficiently large value of k , a tree of size $6k$ can be decomposed into 13 trees; each of size at most k (Lemma 5.4 [12]).

Lemma 1 (Folklore). *Given any tree on n vertices, we can decompose it into two trees (by replicating a single vertex) such that the smaller tree has at most $\lceil \frac{n}{2} \rceil$ vertices and the larger tree has at most $\lceil \frac{2n}{3} \rceil$ vertices.*

3.2 Improvement to Previous Approach: Eligible Trees

An improvement to the analysis in [12] can be achieved by utilising a more refined tree decomposition (than the recursive application of Lemma 1) to provide the approximation guarantee at the final step. To do so, we consider a tree decomposition scheme based on the notion of *eligible trees* as introduced in [3].

Definition 5 ([3]). *Given a directed tree $T = (V_T, E_T)$, an eligible subtree T' is a subtree of T rooted at some vertex $i \in V_T$ such that the forest obtained by deleting the edges with both endpoints in T' , and then all the remaining vertices of degree 0, consists of a single tree.*

Assuming T' is an eligible subtree not identical to T , after deleting all edges with both endpoints in T' , the only vertex of T' with degree strictly greater than 0 is the root vertex of T' . That is, like in Lemma 1, a single vertex is replicated when removing T' from T ; see Fig. 1. The following lemma suggests that, for any tree, there exists an eligible subtree within some specific size range.

Lemma 2 (Lemma 5 [3]). *For each directed tree $T = (V_T, E_T)$, and for each $p \in [1, |V_T|] \cap \mathbb{N}$, there exists an eligible subtree T' of T such that $p/2 \leq |V_{T'}| \leq p$.*

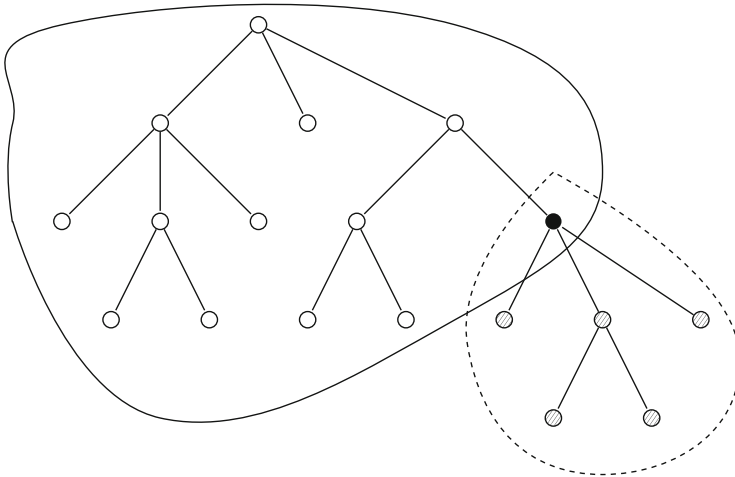


Fig. 1. An example eligible subtree of size 6 (enclosed within the dashed shape). After removing its edges and then all remaining vertices of degree 0 (vertices with lines), a single tree remains (enclosed within the solid shape). A single vertex is replicated in both trees, the black vertex.

We can now proceed to employ the above lemma iteratively toward a decomposition scheme for the tree of size at most $6k$ returned by the Quota Steiner Tree call in Algorithm 2.

Lemma 3. *Let k be an integer. Given any tree T on ak vertices, where $a \in \mathbb{N}$ is a constant, and $k \geq 4a - 2$, we can decompose it into $2a$ subtrees each on at most k vertices.*

Proof. To make T directed, we orient its edges away from some arbitrary vertex picked as the root. Now, we iteratively apply Lemma 2 with $p = k$, until we are left with a tree on at most k vertices.

First, let us show that after i iterations, the remaining tree has at most $ak - i \cdot (k/2 - 1)$ vertices. At the first iteration, there exists an eligible subtree T'_1 such that $k/2 \leq |V_{T'_1}| \leq k$. After removing it from $T_1 := T$ we are left with T_2 of size $|V_{T_1}| - (|V_{T'_1}| - 1)$, since the root of T'_1 remains in T_1 . Hence, $|V_{T_1}| \leq$

$ak - (k/2 - 1)$, since $k/2 \leq |V_{T'_i}|$. Assume that after i iterations of the above procedure, it holds for the remaining tree T_{i+1} that $k < |V_{T_{i+1}}| \leq ak - i \cdot (k/2 - 1)$. We inductively apply Lemma 2 with $p = k$ and get an eligible subtree T'_{i+1} . Removing T'_{i+1} from T_{i+1} , we get T_{i+2} , where $|V_{T_{i+2}}| = |V_{T_{i+1}}| - (|V_{T'_{i+1}}| - 1) \leq ak - i \cdot (k/2 - 1) - (k/2 - 1) = ak - (i + 1) \cdot (k/2 - 1)$.

We proved that, after i removals of eligible subtrees from the original tree, for the remaining tree T_{i+1} it holds $|V_{T_{i+1}}| \leq ak - i \cdot (k/2 - 1)$. For $i = 2a - 1$, we get $|V_{T_{2a}}| \leq ak - (2a - 1) \cdot (k/2 - 1) = ak - ak + 2a + k/2 - 1 = k/2 + 2a - 1$, which is at most k for a sufficiently large value of k , i.e., $k \geq 4a - 2$. Overall, the original tree T_1 has been decomposed into $2a$ trees: $T'_1, T'_2, \dots, T'_{2a-1}$ and T_{2a} , each of which has at most k vertices. \square

Theorem 2. *Algorithm 2 is a $(1 - 1/e)/12$ approximation for BCDS.*

3.3 An Improved Modified Algorithm

In the following proof, we generalize the analysis given in Lemma 5.1 [12] regarding the existence of a greedily selected set (of at most k vertices) with a good intersection to the (neighborhood of the) optimal solution. Below, let D and p refer to the dominating set and profit function returned by GDS (line 1 in Algorithm 2).

Lemma 4. *There exists a set $D' \subseteq D$, $|D'| \leq \lceil ck \rceil$, for some constant $0 < c \leq 1$, such that $p(D') \geq (1 - e^{-c})\text{OPT}$. Furthermore, D' can be connected using at most another $k + \lceil ck \rceil$ Steiner vertices.*

Proof. We define the layers L_1, L_2, L_3 as follows. L_1 contains the (at most k) vertices of an optimal BCDS solution. Let $L_2 = N(L_1)$, meaning that the optimal number of dominated vertices is $\text{OPT} = |L_1 \cup L_2|$. Also, let $L_3 = N(L_2) \setminus L_1$ and $R = V \setminus (L_1 \cup L_2 \cup L_3)$, where R denotes the remaining vertices, i.e., those outside the three layers L_1, L_2, L_3 . Let us now consider the intersection of these layers with the greedy dominating set D returned by GDS (Algorithm 1). Let $L'_i = D \cap L_i$ for $i = 1, 2, 3$ and $D' = \{v_1, v_2, \dots, v_\lambda\}$ denote the first $\lambda = \lceil ck \rceil$ vertices from $L'_1 \cup L'_2 \cup L'_3$ in the order selected by the greedy algorithm. In order to bound the total profit in D' , we define $g_i = \sum_{\mu=1}^i p(v_\mu)$ as the profit we gain from the first i vertices of D' . For the initial value, let $g_0 = 0$.

Proposition 1 (Claim 1 [12]). *For $i = 0, 1, \dots, k - 1$, it holds $g_{i+1} - g_i \geq \frac{1}{k}(\text{OPT} - g_i)$.*

By solving the recurrence in Claim 1, we get $g_i \geq (1 - (1 - \frac{1}{k})^i)\text{OPT}$. Then, for D' , we get $\sum_{v \in D'} p(v) = g_{\lceil ck \rceil} \geq \left(1 - \left(1 - \frac{1}{k}\right)^{\lceil ck \rceil}\right) \text{OPT} \geq \left(1 - \left(1 - \frac{1}{k}\right)^{ck}\right) \text{OPT} \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right)^c \text{OPT} \geq (1 - e^{-c})\text{OPT}$. Moreover, let us show that an extra $k + \lceil ck \rceil$ vertices are enough to ensure that D' is connected. We select a subset $D'' \subseteq L_2$ of size at most $|L_3 \cap D'| \leq \lceil ck \rceil$ to dominate all vertices of $D' \cap L_3$. Then, we ensure that all vertices are connected by simply adding all the k vertices of

L_1 . Thus, $\hat{D} = D' \cup D'' \cup L_1$ induces a connected subgraph that contains at most $k + 2\lceil ck \rceil$ vertices. □

We can now make use of this generalized analysis and suggest a modified algorithm, parameterized by the parameter c , where the Quota Steiner Tree routine is called with a quota of $(1 - e^{-c})\text{OPT}$; see Algorithm 3 below.

Algorithm 3: Modified Greedy Profit Labeling Algorithm for BCDS(c)

Input : A graph $G = (V(G), E(G))$, $k \in \mathbb{N}$
Output: A tree \hat{T} on at most k vertices

- 1 $(D, p) \leftarrow GDS(G)$
- 2 $T \leftarrow QST(G, (1 - e^{-c})\text{OPT}, p)$
- 3 $\hat{T} \leftarrow Best_k(T, p)$
- 4 **return** \hat{T}

Theorem 3. *For some constant $0 < c \leq 1$, there is a $(1 - e^{-c})/(\lceil 8c \rceil + 4)$ approximation for BCDS.*

Proof. By Lemma 4 and Theorem 1, it follows that Algorithm 3 (line 2) returns a tree of size at most $2k + 4\lceil ck \rceil \leq 2k + 4(ck + 1) = (4c + 2)k + 4$ with profit at least $(1 - e^{-c})\text{OPT}$. For a final solution, it suffices to return a subtree of T , namely T' , of size at most k which dominates the maximum number of vertices (call $Best_k(\cdot)$ in line 3 of Algorithm 3). This can be done in polynomial time via dynamic programming; see section 5.2.2 in [12].

To prove a lower bound on the number of vertices T' dominates, we decompose T into a set of subtrees via iteratively removing an eligible tree from T . To do so, we apply Lemma 2 with $p = k$. Like in the proof of Lemma 3, we can prove by induction that after i such removals of eligible subtrees of size at most k , the remaining tree has at most $|T| - i \cdot (k/2 - 1)$ vertices. For $i = \lceil 8c + 3 \rceil$, the remaining tree's size is upper bounded by $(4c + 2)k + 4 - \lceil 8c + 3 \rceil \cdot (k/2 - 1) \leq (4c + 2)k + 4 - (8c + 3) \cdot (k/2 - 1) = k/2 + 8c + 7$, which is at most k for a sufficiently large choice of k , i.e., $k \geq 16c + 14$. Therefore, we can decompose T into $\lceil 8c + 3 \rceil + 1 = \lceil 8c \rceil + 4$ subtrees of size at most k , say $T_1, T_2, \dots, T_{\lceil 8c \rceil + 4}$. Then, from pigeonhole principle and our decomposition, it follows $p(T') \geq \frac{1}{\lceil 8c \rceil + 4} \sum_{i=1}^{\lceil 8c \rceil + 4} p(T_i) \geq \frac{1}{\lceil 8c \rceil + 4} p(T) \geq \frac{1}{\lceil 8c \rceil + 4} (1 - e^{-c})\text{OPT}$. □

For $c = 1$, Theorem 3 matches the approximation ratio already given in Theorem 2. Since the above ratio is a function of the parameter c , we numerically compute its maximum value to $1/11(1 - e^{-7/8})$ attained for $c = 7/8$.

Corollary 1. *There is a $1/11(1 - e^{-7/8})$ -approximation for BCDS.*

3.4 Inapproximability

In this Subsection, we demonstrate a first inapproximability result for BCDS by identifying a reduction from the well known *Maximum Coverage* problem.

Definition 6 (MAX- k -COVER). *Given a positive integer k and a collection of sets $S = \{S_1, S_2, \dots, S_m\}$, find a set $S' \subseteq S$, where $|S'| \leq k$, which maximizes the number of covered elements $|\bigcup_{S_i \in S'} S_i|$.*

Theorem 4 ([5, 11]). *For any $\epsilon > 0$, there is no polynomial time approximation algorithm for MAX- k -COVER within a ratio of $(1 - 1/e + \epsilon)$ unless $P = NP$.*

Let us now demonstrate a *gap-preserving reduction* (Definition 10.2 [1]) which transforms an instance of MAX- k -COVER, namely $MC(S, k)$, where $S = \{S_1, S_2, \dots, S_m\}$ to an instance of BCDS, namely $BCDS(G, k)$, where $G = (V, E)$. For an example illustration, see Fig. 2. For each set $S_i \in S$, we include a vertex s_i in V . Let the union of elements in the set system $\bigcup_{S_i \in S} S_i$ be represented as $\{x_1, x_2, \dots, x_n\}$. For each element x_j , we include q vertices in V , namely $x_{j,1}, x_{j,2}, \dots, x_{j,q}$, where q is a polynomial in m ($q \geq m^2$ suffices). Overall, $|V| = m + qn$. In the edge set E , we include edges (s_i, s_j) , for each $i, j = 1, 2, \dots, m, i \neq j$, and $(s_i, x_{j,z})$, for each i, j such that $x_j \in S_i$ and for each $z = 1, 2, \dots, q$. Notice the size is polynomial in the input of $MC(S, k)$, since we get $|E| \leq \binom{m}{2} + mqn$. In Lemma 5, let $MC(S, k)$, respectively $BCDS(G, k)$, also refer to the optimal solution for the corresponding MAX- k -COVER, resp. BCDS, instance.

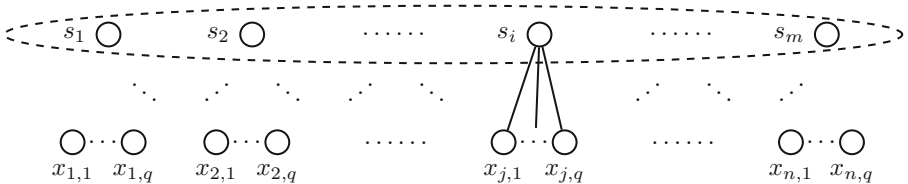


Fig. 2. The graph G constructed for the gap-preserving reduction employed in Lemma 5. Vertices s_i within the dashed ellipse form a clique. Vertex s_i is connected to vertices $x_{j,1}, x_{j,2}, \dots, x_{j,q}$ in G if $S_i \ni x_j$ in $MC(S, k)$.

Lemma 5. *There is a gap-preserving reduction from MAX- k -COVER to BCDS so that,*

- (i) if $MC(S, k) \geq \lambda$, then $BCDS(G, k) \geq \Lambda$, where $\Lambda := m + q\lambda$, and
- (ii) if $MC(S, k) < (1 - \frac{1}{e} + \epsilon) \cdot \lambda$, then $BCDS(G, k) < (1 - \frac{1}{e} + \frac{m}{e(m+q\lambda)} + \epsilon \cdot \frac{q\lambda}{m+q\lambda}) \cdot \Lambda$.

Theorem 5. *For any $\epsilon > 0$, there is no polynomial time approximation algorithm for BCDS within a ratio of $(1 - 1/e + \epsilon)$ unless $P = NP$.*

4 Edge-Vertex Domination

We now turn our attention to edge-vertex domination problems, where the goal is to identify a set of edges which dominate vertices of the graph. We consider both budgeted and partial cover cases.

4.1 Budgeted Edge-Vertex Domination

Let us consider the general case of BEVD (Definition 2), where the selected subset of edges does not need to be connected. We identify a strong connection to the classical MAX- k -COVER problem; see Definition 6 and Theorems 4, 6. On the positive side, in Theorem 7, we prove a $(1 - 1/e)$ -approximation by reducing BEVD to an instance of MAX- k -COVER. On the negative side, we demonstrate a gap-preserving reduction from MAX- k -COVER to BEVD and therefore conclude that the above approximation is the best possible (Theorem 8).

Theorem 6 (Proposition 5.1 [5]). *There exists a $(1 - 1/e)$ -approximation algorithm in polynomial time for MAX- k -COVER.*

Theorem 7. *There exists a $(1 - 1/e)$ -approximation algorithm for BEVD.*

We now proceed and demonstrate a *gap-preserving reduction* (Definition 10.2 [1]) which transforms an instance of MAX- k -COVER, namely $MC(S, k)$, where $S = \{S_1, S_2, \dots, S_m\}$ to an instance of BEVD, namely $BEVD(G, k)$, where $G = (V, E)$. For an illustration, see Fig. 3. The vertex set V contains a “root” vertex v_0 . For each set $S_i \in S$, we include a vertex s_i in V . Let the union of elements in the set system $\bigcup_{S_i \in S} S_i$ be represented as $\{x_1, x_2, \dots, x_n\}$. For each element x_j , we include q vertices in V , namely $x_{j,1}, x_{j,2}, \dots, x_{j,q}$, where q is a polynomial in m ($q \geq m^2$ suffices) Overall, we have $|V| = m + 1 + qn$. In the edge set E , we include the edges (v_0, s_i) , for each $i = 1, 2, \dots, m$, and $(s_i, x_{j,z})$, for each i, j such that $x_j \in S_i$ and for each $z = 1, 2, \dots, q$. The size is polynomial in the input of $MC(S, k)$, since we get $|E| \leq m + mqn$. In Lemma 6, let $MC(S, k)$, respectively $BEVD(G, k)$, refer to the optimal solution for the corresponding max cover, resp. BEVD, instance.

Lemma 6. *There is a gap-preserving reduction from MAX- k -COVER to BEVD so that,*

- (i) *if $MC(S, k) \geq \lambda$, then $BEVD(G, k) \geq \Lambda$, where $\Lambda := m + 1 + q\lambda$, and*
- (ii) *if $MC(S, k) < (1 - \frac{1}{e} + \epsilon) \cdot \lambda$, then $BEVD(G, k) < (1 - \frac{1}{e} + \frac{m+1}{e(m+1+q\lambda)} + \epsilon \frac{q\lambda}{m+1+q\lambda}) \cdot \Lambda$.*

Theorem 8. *For any $\epsilon > 0$, there is no polynomial time approximation algorithm for BEVD within a ratio of $(1 - 1/e + \epsilon)$ unless $P = NP$.*

As a side note, consider the case where the selected edge set is required to be connected. That is, let $BEVD_C$ refer to the budgeted edge-vertex *connected* domination problem. Below, we prove that this problem is equivalent to the budgeted connected dominating set (BCDS) problem researched in Sect. 3.

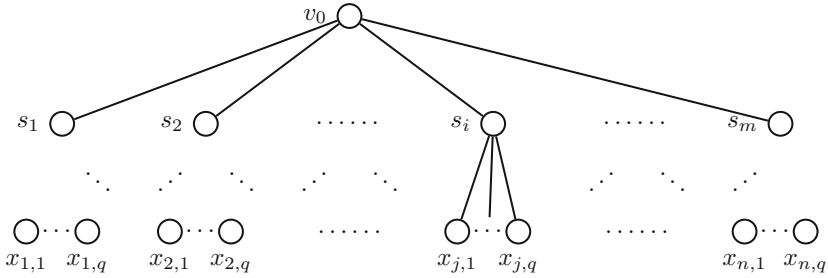


Fig. 3. Graph G constructed for the gap-preserving reduction employed in Lemma 6. Vertex s_i is connected to vertices $x_{j,1}, x_{j,2}, \dots, x_{j,q}$ in G if $S_i \ni x_j$ in $MC(S, k)$.

Proposition 2. For any $G = (V, E)$ where $|V| \geq 2$, and integer $k \geq 2$, a feasible solution S to $BCDS(G, k)$ can be transformed to a solution S_E to $BEVD_C(G, k - 1)$, where $N[S] = N[S_E]$, and vice versa.

4.2 Partial Edge-Vertex Domination

Herein, we prove an $O(\log n)$ -approximation for Partial Edge-Vertex Domination (PEVD); refer to Definition 3. Given a graph $G = (V, E)$ and an integer n' , we need to select a subset $E' \subseteq E$ of minimum size such that it holds $|N[E']| \geq n'$. To approximate the problem, we identify a reduction to *Partial Cover* (PC).

Definition 7 (PARTIAL COVER). Given a universe (set) of elements $X = \{x_1, x_2, \dots, x_n\}$, a collection of subsets of X , $S = \{S_1, S_2, \dots, S_m\}$, and a real $0 < p \leq 1$, find a minimum-size sub-collection of S , say S' , that covers at least a p -part of X , i.e., $|\bigcup_{S_i \in S'} S_i| \geq pn$.

Theorem 9 (Theorems 3, 4 in [24]). PARTIAL COVER is approximable within a factor $\min\{H(\lceil pn \rceil), H(D)\}$, where H is the Harmonic number $H(x) = \sum_{i=1}^x 1/x$ and D is the maximum size of a set in S .

Theorem 10. There exists a $\min\{H(n'), H(2\Delta)\}$ -approximation for PEVD.

5 Conclusion

We propose a new technique to obtain tree decompositions, and a generalized analysis, thus improving the approximation guarantee from $(1 - e^{-1})/13$ to $(1 - e^{-7/8})/11$ for BCDS. Furthermore, we prove a $(1 - 1/e + \epsilon)$ upper bound. Also, we introduce BEVD and PEVD, and provide (tight) approximation bounds.

Regarding future work on BCDS, the goal is to design an algorithm with an improved guarantee. Moreover, it would be interesting to capture the difficulty of the problem with a stronger inapproximability result. We believe that a tight bound lies somewhere between our currently established state of the art.

Related to the edge-vertex case, it would be interesting to consider budgeted and partial versions for other dominating set variants, such as mixed domination [28], where both vertices and edges are selected in order to dominate as many vertices and edges as possible, expansion ratio variants such as in [16], or even eternal domination [15], where a set of guards need to dominate the graph perpetually while moving to protect it against attacks on its vertices.

References

1. Arora, S., Lund, C.: Hardness of approximations. In: *Approximation Algorithms for NP-Hard Problems*, chap. 10, pp. 399–446. PWS Publishing Co., Boston (1997)
2. Bourtrig, R., Chellali, M., Haynes, T.W., Hedetniemi, S.T.: Vertex-edge domination in graphs. *Aequat. Math.* **90**, 355–366 (2016). <https://doi.org/10.1007/s00010-015-0354-2>
3. Bermond, J.-C., et al.: Bin packing with colocations. In: Jansen, K., Mastrolilli, M. (eds.) *WAOA 2016*. LNCS, vol. 10138, pp. 40–51. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51741-4_4
4. Du, D.Z., Wan, P.J.: *Connected Dominating Set: Theory and Applications*. Springer Optimization and its Applications. Springer, New York (2013). <https://doi.org/10.1007/978-1-4614-5242-3>
5. Feige, U.: A threshold of $\ln n$ for approximating set cover. *J. ACM* **45**(4), 634–652 (1998)
6. Garg, N.: Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In: *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC)*, pp. 396–402 (2005)
7. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. *Algorithmica* **20**(4), 374–387 (1998)
8. Horton, J.D., Kilakos, K.: Minimum edge dominating sets. *SIAM J. Discrete Math.* **6**(3), 375–387 (1993)
9. Johnson, D.S., Minkoff, M., Phillips, S.: The prize collecting Steiner tree problem: theory and practice. In: *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pp. 760–769 (2000)
10. Khoa, N.M., Tung, D.D.: Locating fault on transmission line with static var compensator based on phasor measurement unit. *Energies* **11**, 2380 (2018)
11. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. *Inf. Process. Lett.* **70**(1), 39–45 (1999)
12. Khuller, S., Purohit, M., Sarpatwar, K.K.: Analyzing the optimal neighborhood: algorithms for budgeted and partial connected dominating set problems. In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1702–1713 (2014)
13. Khuller, S., Purohit, M., Sarpatwar, K.K.: Analyzing the optimal neighborhood: algorithms for budgeted and partial connected dominating set problems. *SIAM J. Discrete Math.* **34**(1), 251–270 (2020)
14. Kumar, R., Kaur, J.: Efficient beacon placement for network tomography. In: *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pp. 181–186 (2004)
15. Lamprou, I., Martin, R., Schewe, S.: Eternally dominating large grids. *Theoret. Comput. Sci.* **794**, 27–46 (2018)

16. Lamprou, I., Martin, R., Schewe, S., Sigalas, I., Zissimopoulos, V.: Maximum rooted connected expansion. In: 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS), LIPIcs, vol. 117, pp. 25:1–25:14 (2018)
17. Lan, J.K., Chang, G.J.: On the mixed domination problem in graphs. *Theoret. Comput. Sci.* **476**, 84–93 (2013)
18. Lewis, J.: Vertex-edge and edge-vertex parameters in graphs. All Dissertations, 103 (2007)
19. Liu, Y., Liang, W.: Approximate coverage in wireless sensor networks. In: The IEEE Conference on Local Computer Networks 30th Anniversary (LCN 2005), pp. 68–75 (2005)
20. Liu, X., Wang, W., Kim, D., Yang, Z., Tokuta, A., Jiang, Y.: The first constant factor approximation for minimum partial connected dominating set problem in growth-bounded graphs. *Wirel. Netw.* **22**(2), 553–562 (2015). <https://doi.org/10.1007/s11276-015-0981-5>
21. Miyano, E., Ono, H.: Maximum domination problem. In: Proceedings of the Seventeenth Computing: the Australasian Theory Symposium (CATS), vol. 119, pp. 55–62 (2011)
22. Peters, K.W.: Theoretical and algorithmic results on domination and connectivity. Ph.D. thesis, Clemson University, Clemson (1986)
23. Sampathkumar, E., Kamath, S.S.: Mixed domination in graphs. *Sankhya Indian J. Stat.* **54**, 399–402 (1992)
24. Slavík, P.: Improved performance of the greedy algorithm for partial cover. *Inf. Process. Lett.* **64**, 251–254 (1997)
25. Venkatakrisnan, Y.B., Krishnakumari, B.: An improved upper bound of edge-vertex domination number of a tree. *Inf. Process. Lett.* **134**, 14–17 (2018)
26. Wang, B.: Coverage problems in sensor networks. *ACM Comput. Surv.* **43**(4), 32 (2011)
27. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. *SIAM J. Appl. Math.* **38**(3), 364–372 (1980)
28. Zhao, Y., Kang, L., Sohn, M.Y.: The algorithmic complexity of mixed domination in graphs. *Theoret. Comput. Sci.* **412**, 2387–2392 (2011)