Maximum Domination Problem*

Eiji Miyano[1] **Hirotaka Ono**[2]

[1] Department of Systems Design and Informatics, Kyushu Institute of Technology,
Fukuoka 820-8502, Japan.
Email: miyano@ces.kyutech.ac.jp

[2] Department of Economic Engineering, Kyushu University,
Fukuoka 812-8581, Japan.
Email: hirotaka@en.kyushu-u.ac.jp

Abstract

We consider new variants of the vertex/edge domination problems on graphs. A vertex is said to *dominate* itself and its all adjacent vertices, and similarly an edge is said to *dominate* itself and its all adjacent edges. Given an input graph $G = (V, E)$ and an integer k, the k-Vertex (k-Edge) Maximum Domination (k-MaxVD and k-MaxED, respectively) is to find a subset $D_V \subseteq V$ of vertices (resp., $D_E \subseteq E$ of edges) with size at most k that maximizes the cardinality of dominated vertices (resp., edges). In this paper, we first show that a simple greedy strategy achieves an approximation ratio of $(1 - 1/e)$ for both k-MaxVD and k-MaxED. Then, we show that this approximation ratio is the best possible for k-MaxVD unless $\mathcal{P} = \mathcal{NP}$. We also prove that, for any constant $\varepsilon > 0$, there is no polynomial time $1303/1304+\varepsilon$ approximation algorithm for k-MaxED unless $\mathcal{P} = \mathcal{NP}$. However, if k is not larger than the size of the minimum maximal matching, k-MaxED is 3/4-approximable in polynomial time.

keywords: maximum domination, vertex domination, edge domination, approximability, inapproximability.

1 Introduction

1.1 Problem and Motivation

In a graph structure $G = (V, E)$, a vertex u can be considered to *dominate* its adjacent vertices, which is denoted by $N(u)$, and itself by installing some special equipment or facility on u. Similarly, an edge $e = (u, v)$ can be considered to dominate its adjacent edges (i.e., edges defined by $\{(u, i) \mid i \in N(u)\} \cup \{(v, j) \mid j \in N(v)\}$) and itself. The notion of "domination" can be extended to a vertex set or an edge set, which are called a *dominating set* and an *edge-dominating set*, respectively. It is easy to dominate all the vertices or edges if we have a sufficient number of equipments, but we cannot expect such a situation in general. The well-known decision problems (Vertex) Dominating Set (VD, for short) and Edge-Dominating Set (ED, for short) are considered by such a background. Suppose that we are given a graph $G = (V, E)$ and a positive integer k as an input. VD (resp., ED) is requested to answer "yes" if G contains a dominating set $D_V \subseteq V$ (resp., an edge dominating set $D_E \subseteq E$) whose size is at most k, "no" otherwise [GJ79].

To formulate optimization versions of the problems, we can consider several types of objective functions and constraints. A most popular one is to minimize the size of a dominating set (resp., an edge dominating set), which we refer by Minimum Dominating Set (MinVD) (resp., Minimum Edge-Dominating Set (MinED)). Another formulation focuses that from a minimal dominating set, we can construct a spanning forest of a graph, in which each connected component forms a star. Then, the objective function is the sum of the numbers of leaf nodes in the stars. This formulation is called Spanning Star Forest problem, which has an application in bioinformatics fields [NSH+07].

In this paper, we consider yet another natural optimization version: Given a positive integer k, we would like to find a subset D_V of vertices (resp., D_E of edges) with size k that maximizes the cardinality of dominated vertices (resp., edges), that is, $\bigcup_{v \in D_V} N(v) \cup D_V$ (resp., $\bigcup_{(u,v) \in D_E} (\{(u, v)\} \cup \{(u, i) \mid i \in N(u)\} \cup \{(v, i) \mid i \in N(v)\})$). We call the vertex version of the problem k-Vertex Maximum Domination (k-MaxVD, for short) and the edge version k-Edge Maximum Domination (k-MaxED, for short). Note that, in k-MaxVD (resp., k-MaxED), a solution is not always a vertex (resp., an edge) dominating set in the ordinary sense, and they dominate only a part of the whole vertices (resp., edges) in general. To emphasize this, we sometimes use the terms *partial (vertex) dominating set* and *partial edge dominating set* to refer solutions of k-MaxVD and k-MaxED.

Since VD and ED are known to be $\mathcal{NP}$-complete, all of these optimization versions of the problems trivially turn out to be $\mathcal{NP}$-hard. However, since the approximability heavily depends on the objective functions, the upper/lower bounds of the approximation ratio are not obvious; in this paper, we focus on the k-MaxVD and k-MaxED from the viewpoint of (in)approximability.

1.2 Our contribution

In this paper, we consider (in)approximability of k-MaxVD and k-MaxED, and obtain the following results: For k-MaxVD, there is a polynomial time $(1 - 1/e)$-approximation algorithm, but no polynomial time constant factor approximation algorithm whose factor is better than $(1 - 1/e)$ unless $\mathcal{P} = \mathcal{NP}$. That is, the optimal bound for the approximability is obtained under $\mathcal{P} \neq \mathcal{NP}$. Since k-MaxVD forms a special case of Maximum Coverage Problem (MaxCP), the same approximability bound is straightforwardly obtained, but the inapproximability bound is not. Nevertheless, we can show the same inapproximability bound as MaxCP is obtained via a gap-

*This work is partially supported by KAKENHI (No.20500017, 21680001 and 22650004).

preserving reduction.

For k-MaxED, we show that there is also a polynomial time $(1-1/e)$-approximation algorithm, but even if the maximum degree of input graphs is at most 7, there is no polynomial time approximation algorithm with any constant factor better than 1303/1304 for k-MaxED, unless $\mathcal{P}=\mathcal{NP}$. The approximability can be improved into 3/4 in the case when k is at most the size of minimum maximal matching of the graph, which would be a natural setting for the k-MaxED problem. In the results, although the $(1-1/e)$-approximability is also based on the result of MaxCP, the other is different; the 3/4-approximability is based on a deterministic rounding for a relaxed semi-definite programming of k-MaxED.

1.3 Related Work

In graph theory, the notion of dominating/covering vertices/edges of graphs has been extensively studied (see, e.g., [Har69, HHS98, Hoc97]). The minimization versions of dominating problems, MinVD and MinED, are the classical $\mathcal{NP}$-hard optimization problems, and thus there is much of literature studying their (in)approximability: Since MinVD is a special instance of Minimum Set Cover, it is approximable within $1+\log n$ for an n-vertex graph. On the other hand, unless $\mathcal{P}=\mathcal{NP}$, no polynomial-time algorithm can generally provide a $(1-\varepsilon)\log n$ approximation guarantee for any ε [Fei98]. As for MinED, there is a polynomial-time 2-approximation algorithm: One can see that any maximal matching of a graph is one possible edge dominating set of the graph. Furthermore, the smallest maximal matching has the same size as the smallest edge dominating set [YG80]. Thus, all we have to do is just find a maximal matching in polynomial time since it can be at most two times as large as the smallest maximal matching. This 2-approximation algorithm is quite simple but it is currently the best one. A general inapproximability result was given by Chlebík and Chlebíková [CC06b], who proved that it is $\mathcal{NP}$-hard to approximate MinED within 7/6. They also showed that it is $\mathcal{NP}$-hard to approximate MinED within $(7+\Delta/n)/(6+\Delta/n)$ for graphs with maximum degree at most Δ.

The maximization versions are also the fundamental optimization problems and furthermore they have many real-life applications rising in facility location and layout problems, such as service centers, emergency vehicles, and wireless access points (e.g., see [CR74, Hoc97, MZH83] and the references therein for application examples). Also, for example, the biological data analysis such as the analysis of differential gene expression data [LLP+04] and the selection of maximally informative set of SNPs (single-nucleotide polymorphisms) [AYN+05, WJ08] can be essentially formulated by the maximum dominating/covering vertex/edge problems. However, the maximum versions have not been well studied yet; there are only few results on the (in)approximability: The k-Vertex Maximum Cover (k-MaxVC) was first introduced by Petrank [Pet94], who proved that there is no polynomial time $(1-\varepsilon)$ factor approximation algorithm for k-MaxVC unless $\mathcal{P}=\mathcal{NP}$ for a small positive ε. (Note that, however, the specific value of ε has not been proved so far; in this paper we shall show that $\varepsilon \geq 1/163$ holds even if the maximum degree is at most 3 in Section 4, which is an important by-product.) Ageev and Sviridenko proved [AS04] that there is a 3/4 approximation algorithm for k-MaxVC, based on an LP relaxation. Then, Han et al. [HYZZ02] showed that SDP-based algorithms achieve improved approximation ratios for some restricted k. As far as the authors know, however, there is no result for k-MaxVD and k-MaxED so far. (Remark that Budgeted Maximum Graph Coverage introduced in [KMPRW02] has a great resemblance to k-MaxVD and k-MaxED, but it must be quite difficult to formulate our problems in the framework of Budgeted Maximum Graph Coverage.)

1.4 Organization

In Section 2 we present a simple greedy algorithm of approximation ratio $(1-1/e)$ both for k-MaxVD and k-MaxED. In Section 3, we prove that it is $\mathcal{NP}$-hard to approximate k-MaxVD within ratio $(1-1/e)$. Section 4 discusses the inapproximability of k-MaxED; we prove that it is $\mathcal{NP}$-hard to approximate k-MaxED within ratio 1303/1304 even if the maximum degree of an input graph is at most 7. Finally, a polynomial time 3/4 factor approximation algorithm for a typical class of k-MaxED is presented in Section 5.

2 Approximability of Maximum Coverage Problem

Both k-MaxVD and k-MaxED can be regarded as special cases of Maximum Coverage Problem (MaxCP), which is formalized as follows:

MaxCP:

Instance: A collection of sets $\mathcal{S}=\{S_1,S_2,\cdots,S_m\}$ defined over a domain of elements $X=\{x_1,x_2,\cdots,x_n\}$, a weight function $w: X \to Q$ where Q is the set of nonnegative rational numbers, and a positive integer $\ell \leq |\mathcal{S}|$.

Goal: Find a collection of sets $\mathcal{S}' \subseteq \mathcal{S}$ whose size is at most ℓ such that the total weight of elements covered by $\mathcal{S}'$, i.e., $\sum_{x_i \in X(\mathcal{S}')} w(x_i)$, where $X(\mathcal{S}')=\bigcup_{S_i \in \mathcal{S}'} S_i$, is maximized.

For the MaxCP, a greedy heuristic is naturally considered:

Algorithm Greedy

Step 1. Let $\mathcal{S}'=\emptyset$.

Step 2. For each $S \in \mathcal{S} \setminus \mathcal{S}'$, compute $f(S,\mathcal{S}')=\sum_{x_i \in S \setminus X(\mathcal{S}')} w(x_i)$.

Step 3. Let $S^*=argmax_{S \in \mathcal{S} \setminus \mathcal{S}'}\{f(S,\mathcal{S}')\}$, and let $\mathcal{S}' := \mathcal{S}' \cup \{S^*\}$. If $|\mathcal{S}'|=\ell$, output $\mathcal{S}'$ and exit. Otherwise, go to Step 2.

The greedy algorithm obviously runs in polynomial time. As for the approximation performance, the following theorem is known.

Theorem 1 ([Hoc97]) *The greedy algorithm achieves an approximation factor of* $(1-1/e)$ *for* MaxCP. $\square$

As mentioned above, both k-MaxVD and k-MaxED are special cases of MaxCP: For an instance of k-MaxVD, we can construct an instance of MaxCP by considering $X=V$ and $\ell=k$, and defining $\mathcal{S}$ by $S_i=N(v_i)\cup\{v_i\}$ and $w(v_i)=1$ for all $i=1,\ldots,n$. Also for an instance of k-MaxED, we can construct an instance of MaxCP by considering $X=E$ and $\ell=k$, and defining $\mathcal{S}$ by $S_{uv}=\{(u,i) \mid i \in N(u)\}\cup\{(v,i) \mid i \in N(v)\}\cup\{(u,v)\}$ and $w((u,v))=1$ for all $(u,v)\in E$. Thus, both k-MaxVD and k-MaxED have polynomial time $(1-1/e)$-approximation algorithms.

As for the inapproximability of MaxCP, the following result is known.

Theorem 2 ([Fei98]) *No polynomial time approximation algorithm with a constant approximation ratio better than $(1-1/e)$ exists for* MaxCP *unless* $\mathcal{P}=\mathcal{NP}$, *even if the weights of elements are restricted to* 1. $\square$

By Theorems 1 and 2, the above simple greedy heuristic algorithm for MaxCP is optimal in the sense of approximability under $\mathcal{P}\neq\mathcal{NP}$. However, this does not immediately imply the optimality of the greedy algorithm for k-MaxVD and k-MaxED.

3 Inapproximability of k-MaxVD

In this section, we discuss the hardness of approximation of k-MaxVD. As seen above, an instance of k-MaxVD is easily formulated as an instance of MaxCP, but the opposite direction is not trivial. Its difficulty is in the restriction that the covering objects and the covered objects are from an identical set in k-MaxVD.

However, we can still show that k-MaxVD has the same bound on inapproximability under $\mathcal{P}\neq\mathcal{NP}$. The bound is shown via a *gap-preserving reduction* [AL95] from the variant of MaxCP whose weight is uniform (MaxUCP, for short), which also has the inapproximability bound $(1-1/e)$.

Let $OPT_{mcp}(I)$ denote the weight of elements covered by a collection of sets output by an optimal algorithm for the instance I of MaxUCP. Also, let $OPT_{mvd}(G)$ be the weight of the dominated vertices of output by an optimal algorithm for graph G of k-MaxVD.

Lemma 3 *There is a gap-preserving reduction from* MaxUCP *to* k-MaxVD *that transforms an instance* I *of* MaxUCP *to a graph* $G=(V,E)$ *of* k-MaxVD *such that*

(i) *if* $OPT_{mcp}(I)=\texttt{max}$, *then* $OPT_{mvd}(G)\geq\ell(q+1)+p\cdot\texttt{max}$, *and*

(ii) *if* $OPT_{mcp}(I)\leq(1-\frac{1}{e})\texttt{max}$, *then* $OPT_{mvd}(G)\leq(1-\frac{1}{e}+\frac{\ell(q+1)}{e(\ell(q+1)+p\cdot\texttt{max})})(\ell(q+1)+p\cdot\texttt{max})$, *where* p *and* q *are any positive integer larger than* $|I|$.

Proof: Consider an instance I of MaxUCP; a collection of sets $S=\{S_1,S_2,\cdots,S_m\}$ defined over a domain of elements $X=\{x_1,x_2,\cdots,x_n\}$, and a positive integer ℓ. Then, we construct the following graph, illustrated in Figure 1. Let $V_R=\{r_{1,1},r_{1,2},\ \cdots,r_{1,q},r_{2,1},r_{2,2},\ \cdots,r_{m,q-1},r_{m,q}\}$ be a set of mq nodes associated with $|S|=m$. Also, let $V_S=\{\alpha_1,\alpha_2,\cdots,\ \alpha_m\}$ be a set of m nodes corresponding to the m sets, S_1 through S_m, and $V_X=\{\beta_{1,1},\beta_{1,2},\ldots,\beta_{1,p},\beta_{2,1},\ldots,\beta_{n,p}\}$ be a set of np nodes, in which $\beta_{i,1},\ldots,\beta_{i,p}$ correspond to x_i. These V_R, V_S, and V_X are pairwise disjoint, and let $V=V_R\cup V_S\cup V_X$. The set E of edges is defined as follows: $E_1=\{(\alpha_i,r_{i,1}),(\alpha_i,r_{i,2}),\cdots,(\alpha_i,r_{i,q})\mid i=1,2,\cdots,m\}$, $E_2=\{(\alpha_i,\beta_{j,1}),\ldots,(\alpha_i,\beta_{j,p})\mid x_j\in S_i$ for each i and each $j\}$, and $E=E_1\cup E_2$. Finally, we set $k=\ell$. This reduction can be done in polynomial time if both p and q are bounded by some polynomial of $|I|$.

(i) Suppose that $OPT_{mcp}(I)=\texttt{max}$ and the optimal collection of sets is $OPT=\{S_{i_1},S_{i_2},\cdots,S_{i_\ell}\}$ where $\{i_1,i_2,\cdots,i_\ell\}\subseteq\{1,2,\cdots,m\}$. Then, if we select a set $D=\{\alpha_{i_1},\alpha_{i_2},\cdots,\alpha_{i_k}\}$ of k $(=\ell)$ nodes as a partial dominating set, then we can obtain the dominated set with size $k(q+1)+p\cdot\texttt{max}$, since the nodes α_{i_j}'s are connected with exactly $p\cdot\texttt{max}$ β's and each α_{i_j} is connected with exactly q r's.

(ii) We show this by contradiction. We first suppose that $OPT_{mcp}(I)\leq(1-\frac{1}{e})\texttt{max}$ but $OPT_{mvd}(G)>(1-\frac{1}{e}+\frac{\ell(q+1)}{e(\ell(q+1)+p\cdot\texttt{max})})(\ell(q+1)+p\cdot\texttt{max})=(1-\frac{1}{e})p\cdot\texttt{max}+\ell(q+1)$ holds for some p and q with $p\geq|I|$ and $q\geq|I|$. Let D be an optimal partial dominating set of G. We partition D into $D_R=D\cap V_R$, $D_S=D\cap V_S$ and $D_X=D\cap V_X$. We then estimate the number of dominated vertices. We can see that a vertex in D_R can dominate only itself and its adjacent V_S vertex, that is, at most 2 vertices. Also a vertex $\beta_{j,a}$ in D_X can dominate itself and vertices α_i satisfying $x_j\in S_i$, that is, at most $m+1$ vertices. Thus $D_R\cup D_X$ can dominate at most $2|D_R|+(m+1)|D_X|$. As for D_S, it can be considered a subset of $\{S_1,S_2,\ldots,S_m\}$ in I, which covers some elements of X. Let s denote the number of the covered elements by D_S. Then D_S can dominate at most $(q+1)|D_S|+ps$ vertices. Thus D can dominate at most $ps+2|D_R|+(m+1)|D_X|+(q+1)|D_S|$ vertices, where $|D_R|+|D_X|+|D_S|=\ell$. It follows that $ps+2|D_R|+(m+1)|D_X|+(q+1)|D_S|\geq OPT_{mvd}(G)>(1-\frac{1}{e})p\cdot\texttt{max}+\ell(q+1)$. Since $\ell(q+1)=(|D_R|+|D_X|+|D_S|)(q+1)>2|D_R|+(m+1)|D_X|+(q+1)|D_S|$, we have $s>(1-\frac{1}{e})\cdot\texttt{max}$, which contradicts $OPT_{mcp}(I)\leq(1-\frac{1}{e})\cdot\texttt{max}$. $\square$

Note that this proof holds also in the case when $q=m$. By taking p large enough but in polynomial of $|I|$ (e.g., $p=|I|^2$), we obtain the following theorem.

Theorem 4 *There is no polynomial time approximation algorithm with any constant factor better than* $(1-\frac{1}{e})$ *for* k-MaxVD *unless* $\mathcal{P}=\mathcal{NP}$. $\square$

4 Inapproximability of k-MaxED

4.1 Degree Bounded Graphs

The goal of this section is to show that the hardness of approximation for the edge version k-MaxED via a *gap-preserving reduction* [AL95]. Our first hardness result is proved by the reduction from k-Maximum Vertex Cover with Bounded Degree (k-MaxVC(Δ)) problem, and holds even for graphs with maximum degree bounded by $2\Delta+1$:

k-**MaxVC(Δ):**

Instance: A graph $G=(V,E)$ with maximum degree Δ, where $V=\{v_1,v_2,\ldots,v_n\}$ is the set of vertices and $E=\{e_1,e_2,\ldots,e_m\}\subseteq V\times V$ is the set of edges, and a positive integer $\ell\leq m$.

Goal: Find $V'\subseteq V$ with $|V'|=\ell$ such that $|\{(u,v)\mid u\in V'\text{ or }v\in V'\}|$ is maximized.

Theorem 5 ([Pet94]) k-MaxVC(Δ) *is* $\mathcal{APX}$*-hard for* $\Delta\geq3$, *that is, there exists a constant* ε_Δ *such that* k-MaxVC(Δ) *has no polynomial time approximation algorithm whose approximation factor is better than* $1-\varepsilon_\Delta$ *unless* $\mathcal{P}\neq\mathcal{NP}$. $\square$

It is important to note that the hardness parameter ε_Δ of approximation comes from the gap-preserving *identity* reduction from Minimum Vertex Cover with Bound Degree (MinVC(Δ)), and the hardness result of approximation for MinVC(Δ) shown in [PY91]. The reduction implies that if an algorithm ALG for k-MaxVC can find a vertex cover with cardinality k of G that covers at least $(1-\varepsilon_\Delta)|E|$ edges in G, then ALG can also find a cover of all edges

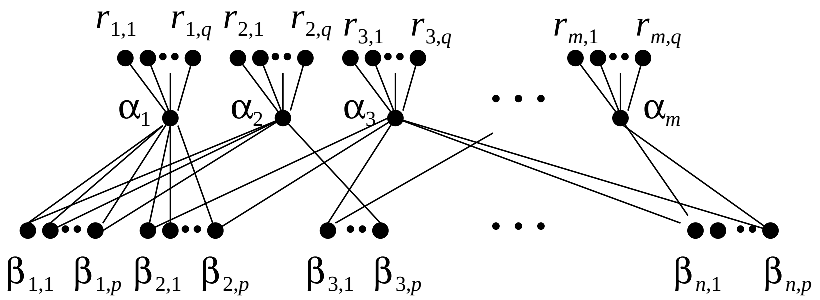

Figure 1: Constructed graph by the gap-preserving reduction from MaxUCP to k-MaxVD

with at most $k+\varepsilon_\Delta|E|$ vertices for MinVC. Thus, the approximation ratio of ALG for MinVC is obtained as follows:

$$\frac{\texttt{ALG}}{\texttt{OPT}} \le 1+\frac{\varepsilon_\Delta|E|}{k} \le 1+\frac{\varepsilon_\Delta|V|\Delta}{2k},$$

where OPT denotes an optimal algorithm for MinVC. The second inequality comes from $|E| \le |V|\cdot\Delta/2$. By the lower bound 100/99 of approximation for MinVC(3) shown by Chlebík and Chlebíková [CC06a] and careful reading of their proofs, the following two inequalities can be obtained:

$$\varepsilon_3 \ge \frac{2}{297}\cdot\frac{k}{|V|}, \quad \text{and} \quad k \ge \frac{958}{1048}\cdot|V|.$$

As a result, the approximation hardness in Theorem 5 is satisfied the following:

$$1-\varepsilon_3 \le \frac{309340}{311256} \le 0.993845 \le \frac{162}{163}.$$

Theorem 6 ([Pet94, CC06a]) *There is no polynomial time approximation algorithm with any constant factor better than* 162/163 *for* MinVC(3) *unless* $\mathcal{P}=\mathcal{NP}$. □

Now we consider the gap-preserving reduction from MinVC(Δ), which constructs a graph with maximum degree $2\Delta+1$ of, say, k-MaxED($2\Delta+1$) from a graph with maximum degree Δ of MinVC(Δ). Let $OPT_{vc}(G)$ denote the size of an optimal solution of an instance I of MinVC(Δ), and $OPT_{med}(G)$ be the size of an optimal solution for G of k-MaxED.

Lemma 7 *There is a reduction from* MinVC(Δ) *to* k-MaxED *that transforms an instance* $G=(V,E)$ *of* MinVC(Δ) *to a graph* $G'=(V',E')$ *of* k-MaxED *such that*

(i) *if* $OPT_{vc}(G)=\texttt{max}$, *then* $OPT_{med}(G') \ge \ell(2\Delta+1)+\texttt{max}$,

(ii) *if* $OPT_{vc}(G) \le c\cdot\texttt{max}$ *where* $c\le 1$, *then* $OPT_{med}(G') \le \frac{c+2\Delta+1}{2\Delta+2}(\ell(2\Delta+1)+\texttt{max})$, *and*

(iii) *the maximum degree of* G' *is at most* $2\Delta+1$.

Proof: Consider an instance G of MinVC(Δ), a vertex set $V=\{v_1,v_2,\ldots,v_n\}$, an edge set $E=\{e_1,e_2,\ldots,e_m\}\subseteq V\times V$, and a positive integer ℓ. Then, we construct the following graph $G'=(V',E')$, where $V'=V\cup V_C\cup V_L$ and $E'=E\cup E_C\cup E_L$ are defined by $V_C=\{v'_i \mid v_i\in V\}$, $V_L=\{u_{i,1}\ldots,u_{i,2\Delta} \mid i=1,\ldots,n\}$, $E_C=\{(v_i,v'_i)\mid i=1,\ldots,n\}$ and $E_L=\{(v'_i,u_{i,1}),\ldots,(v'_i,u_{i,2\Delta})\mid i=1,\ldots,n\}$. This reduction can be done in polynomial time.

(i) Suppose that $OPT_{vc}(G)=\texttt{max}$ and the optimal subset of vertices is $V^*=\{v_{1^*},v_{2^*},\cdots,v_{\ell^*}\}$ where $\{1^*,2^*,\cdots,\ell^*\}\subseteq\{1,2,\cdots,n\}$. Then, we select an edge set $D=\{(v_{1^*},v'_{1^*}),(v_{2^*},v'_{2^*}),\ldots,(v_{\ell^*},v'_{\ell^*})\}$ of ℓ edges as a partial edge dominating set. One can see that D dominates $\ell(2\Delta+1)+\texttt{max}$ edges since the nodes v_{i^*}'s are connected with exactly max edges in E and each $v'_{i^*}$is connected with exactly 2Δ u's.

(ii) We show this by contradiction. We first suppose that $OPT_{vc}(G)\le c\cdot\texttt{max}$ but $OPT_{med}(G') > \frac{c+2\Delta+1}{2\Delta+2}(\ell(2\Delta+1)+\texttt{max})$ holds. Let D be an optimal partial edge dominating set of G. We partition D into $D_{org}=D\cap E$, $D_C=D\cap E_C$ and $D_L=D\cap E_L$. We then estimate the number of dominated edges. We can see that an edge $(v'_i,u_{i,j})$ in D_L dominates itself, the other edges $(v'_i,u_{i,j'})$ and (v_i,v'_i), that is, exactly $2\Delta+1$ edges. An edge (v_i,v_j) in D_{org} can dominate itself, its adjacent edges in E, that is, at most $2\Delta+1$ edges. Thus $D_L\cup D_{org}$ can dominate at most $(2\Delta+1)(|D_L|+|D_{org}|)$. As for D_C, it can be considered a subset of V, which covers some edges in E. Let s denote the number of the covered edges by D_C. Then D_C can dominate at most $(2\Delta+1)|D_C|+s$ edges, because each edge (v_i,v'_i) in E_C dominates itself and its adjacent $(v'_i,u_{i,j})$'s. Thus D can dominate at most $(2\Delta+1)(|D_L|+|D_{org}|+|D_C|)+s=\ell(2\Delta+1)+s$ edges. It follows that

$$\begin{aligned}\ell(2\Delta+1)+s &\ge OPT_{med}(G')\\ &> \frac{c+2\Delta+1}{2\Delta+2}(\ell(2\Delta+1)+\texttt{max}).\end{aligned}$$

Since

$$\begin{aligned}&(c+2\Delta+1)(\ell(2\Delta+1)+\texttt{max})-\\ &\qquad(2\Delta+2)(\ell(2\Delta+1)+c\cdot\texttt{max})\\ =\;&(2\Delta+1)(1-c)(\texttt{max}-\ell)\\ \ge\;&0,\end{aligned}$$

(the inequality holds by $\texttt{max}\ge\ell$ and $c\le 1$), we have $\ell(2\Delta+1)+s > \ell(2\Delta+1)+c\cdot\texttt{max}$, which contradicts $OPT_{vc}(G)\le c\cdot\texttt{max}$. □

By this lemma, we can show the following hardness of approximation by the above reduction:

Theorem 8 *Let* $c(\Delta)$ *be a lower bound of the approximation factor for* MinVC(Δ), *that is, there is no polynomial time approximation algorithm with any constant factor better than* $c(\Delta)$ *unless* $\mathcal{P}=\mathcal{NP}$. *Then, there is no polynomial time approximation algorithm with any constant factor better than* $\frac{c(\Delta)+2\Delta+1}{2\Delta+2}$ *for* k-MaxED *unless* $\mathcal{P}=\mathcal{NP}$. □

By setting $\Delta=3$ and $c(\Delta)\le 162/163$, the following inapproximability for the bounded degree case of k-MaxED can be obtained:

Corollary 9 *There is no polynomial time approximation algorithm with any constant factor better than* $1303/1304$ *for* k-MaxED *even if the maximum degree of an input graph is at most* 7*, unless* $\mathcal{P} = \mathcal{NP}$. □

5 Approximability of k-MaxED

In this section, we present an approximation algorithm for k-MaxED that works in the case when k is at most the size of minimum edge dominating set, which is a natural setting for considering k-MaxED. The approximation factor of the algorithm is $3/4 = 0.75$, which improves the usual one, $(1 - 1/e) \leq 0.6322$. Our algorithm is based on a technique called *pipage rounding*, which is proposed for an LP-relaxation algorithm of MaxCP by Ageev and Sviridenko [AS04].

Here we give a short introduction of the pipage rounding. Suppose that the target problem is formulated as an n-dimensional binary mathematical programming in two ways; the two mathematical programming formulations have the same constraints and different objective functions. The pipage rounding is a rounding scheme that well utilizes the two objective functions $F(x)$ and $L(x)$. The values of $F(x)$ and $L(x)$ should coincide at each binary point. $L(x)$ gives a relaxed optimal solution in the sense that a relaxed version (typically, an LP-relaxation) of the mathematical programming with objective function $L(x)$ should be solved efficiently (i.e., in polynomial time), and $F(x)$ gives a criteria for good rounding. $F(x)$ and $L(x)$ need to satisfy mainly two properties. (A) One is that, for some $c > 0$, $F(x) \geq cL(x)$ holds for any $x \in [0, 1]^n$, which is used for giving an approximation ratio. (B) The other is a certain convexity (monotonicity) of $F(x)$; based on this, we can sequentially round a fractional (optimal) solution of the mathematical programming of $L(x)$ so that $F(x)$ value of the resulting solutions monotonically increases.

The algorithm works as follows: First, we solve a relaxed version of the mathematical programming with objective function $L(x)$ in polynomial time. We obtain a fractional optimal solution x^* whose value is $L(x^*)$. We then round the fractional solution into a binary solution based on the function values of $F(x)$. By the convexity (monotonicity), the resulting binary solution x' satisfies that $F(x') \geq F(x^*) \geq cL(x^*)$. This implies that the algorithm is a polynomial time c-approximation algorithm. For more detail, see [AS04].

Now we design an approximation algorithm based on the pipage rounding, but the domain of the rounding is different from the ordinary one. In the ordinary one, the domain in which both (A) and (B) hold is all $x \in [0, 1]^n$, whereas the domain of our algorithm is more restricted. We formulate k-MaxED as the following binary program:

$$\begin{aligned} \max \quad & F(x) & (1)\\ \text{subject to} \quad & \sum_{e \in E} x_e = k & (2)\\ & x_e \in \{0, 1\},\ e \in E, & (3) \end{aligned}$$

where $F(x) = \sum_{e=(u,v) \in E} w_e(1 - \prod_{i \in N(u)}(1 - x_{(u,i)}) \prod_{i \in N(v)}(1 - x_{(v,i)}))$. By considering $x_e = 1$ if $e \in D_E$ and $x_e = 0$ otherwise, it is easy to see that the optimal solutions of the problem defined by (1) - (3) correspond to the optimal sets of k-MaxED. Also, we consider another formulation of k-MaxED by adopting the objective function $L(x)$, where $L(x) = \sum_{e=(u,v) \in E} w_e \times \min\{1, \sum_{(u,i) \in N(u)} x_{(u,i)} + \sum_{(v,i) \in N(u)} x_{(v,i)} - x_{(u,v)}\}$. It is also easy to see that the optimal solutions of the problem defined by objective function $L(x)$ and constraints (2) and (3) correspond to the optimal sets of k-MaxED.

Here, we consider the case when k is at most the size of the minimum edge dominating set, which is equal to the minimal maximal matching. This implies that the optimal sets of k-MaxED forms matchings, otherwise we can increase the value of solutions. Thus even if we add the following constraint (4), the optimal solutions still correspond to the optimal sets of k-MaxED:

$$x_e x_{e'} = 0, \qquad \forall v \in V, \forall e, e' \in \Gamma(v), \tag{4}$$

where $\Gamma(v) = \{(u, v) \mid u \in N(v)\}$, i.e., the set of edges connected to vertex v. We call this constraint (4) *matching constraint*.

Here we consider the mathematical programming that maximizes $L(x)$ under (2), (4) and a relaxed (3), i.e., $x_e \in [0, 1]$ for $e \in E$. This relaxed problem, say, (RP) is not a linear programming due to (4). Instead, (RP) can be transformed into a semidefinite programming, which can be solved within an additive error of ε in polynomial time, for a given any $\varepsilon > 0$ (ε is a part of the input, so the running time dependence on ε is polynomial in $\log 1/\varepsilon$). This can be done through the ellipsoid algorithm [GLS88]. Here is an SDP-formulation of the problem (SDP).

$$\begin{aligned} \max \quad & L'(x) \stackrel{\text{def}}{=} \sum_{e \in E} w_e \frac{z_e \cdot y_0 + 1}{2} & (5)\\ \text{subject to} \quad & \sum_{e \in E} y_e \cdot y_0 = 2k - m, & (6)\\ & z_e \cdot y_0 \leq \sum_{e' \in \Gamma(i) \cup \Gamma(j)} y_{e'} \cdot y_0, & \\ & \qquad\qquad \forall e = (i, j) \in E & (7)\\ & (y_e + y_0) \cdot (y_{e'} + y_0) = 0, & (8)\\ & y_e, z_e \in B, \qquad \forall e \in E & (9)\\ & y_0 \in B, & (10) \end{aligned}$$

where B is the unit sphere in R^{2m+1}, that is, $B = \{y \in R^{2m+1} \mid ||y||_2 = 1\}$. Namely, y_e's and z_e's are unit vectors in the R^{2m+1} space. In (SDP) and (RP), (6) and (4) correspond to (2) and (8), respectively. Note that z_e in (7) (and (5)) is introduced to represent min in $L(x)$. After solving (SDP) in polynomial time, we obtain an optimal solution y^*. Then it is easy to see that we can obtain an optimal solution x^* of (RP) by setting

$$x_e = \frac{1 + y_e^* \cdot y_0^*}{2}.$$

Due to the matching constraint, in an optimal solution $x^* = (x_1^*, x_2^*, \ldots, x_m^*)$ of (RP), $E^* = \{e \mid x_e^* > 0\}$ forms a matching of G.

We now show that the functions $F(x)$ and $L(x)$ satisfy properties (A'): there exists $c > 0$ such that $F(x) \geq cL(x)$ for each $x = (x_1, x_2, \ldots, x_m) \in [0, 1]^E$ satisfying that $\{e \mid x_e > 0\}$ is a matching of G, and (B): the function $\phi(\varepsilon, x, i, j) = F(x_1, \ldots, x_i + \varepsilon, \ldots, x_j - \varepsilon, \ldots, x_n)$ is convex with respect to $\varepsilon \in [-\min\{x_i, 1 - x_j\}, \min\{1 - x_i, x_j\}]$ for each pair of indices i and j and each $x \in [0, 1]^E$.

We first show that property (A) holds with $c = 3/4$. We consider $F(x)$ and $L(x)$ of x satisfying that $\{e \mid x_e > 0\}$ is a matching of G. If $x_e = x_{(u,v)} > 0$, the corresponding term in $F(x)$ is

$$w_e \times (1 - \prod_{i \in N(u)} (1 - x_{(u,i)}) \prod_{i \in N(v)} (1 - x_{(v,i)})) = w_e x_e,$$

and the corresponding term in $L(x)$ is

$$w_e \times \min\{1, \sum_{(u,i)\in N(u)} x_{(u,i)} + \sum_{(v,i)\in N(u)} x_{(v,i)} - x_{(u,v)}\} = w_e x_e.$$

Otherwise, (i.e., $x_e = x_{(u,v)} = 0$), edge e is dominated by at most two edges, say, $x_{(u,p)}$ and $x_{(v,q)}$. Then the corresponding term in $F(x)$ is

$$\begin{aligned} & w_e \times (1 - \prod_{i\in N(u)} (1 - x_{(u,i)}) \prod_{i\in N(v)} (1 - x_{(v,i)})) \\ = \quad & w_e(1 - (1 - x_{(u,p)})(1 - x_{(v,q)})), \end{aligned}$$

and the corresponding term in $L(x)$ is

$$w_e \times \min\{1, \sum_{(u,i)\in N(u)} x_{(u,i)} + \sum_{(v,i)\in N(u)} x_{(v,i)} - x_{(u,v)}\} = w_e \times \min\{1, (x_{(u,p)} + x_{(v,q)})\}.$$

Since it is known that the following inequality

$$1 - \prod_{i}^{r}(1 - y_i) \geq (1 - (1 - 1/r)^r) \min\{1, \sum_{i=1}^{r} y_i\},$$

is valid for all $0 \leq y_i \leq 1, i = 1, \ldots, r$ (for the proof, see [GW94]), $F(x) \geq (1-(1-1/2)^2)L(x) = 3/4L(x)$ holds for any x satisfying that $\{e \mid x_e > 0\}$ is a matching of G.

It is easy to see that property (B) holds, because $\phi(\varepsilon, x, i, j)$ is a quadratic polynomial in ε, which is convex.

By these, the concrete algorithm is described as follows:

Algorithm PIPAGE k-MAXED

Step 1. Solve (RP) in polynomial time, and let x^* be the optimal solution. Let $x := x^*$.

Step 2. If x is a binary vector, output x as x', and exit.

Step 3. Find two different components x_i and x_j that are strictly larger than 0 and smaller than 1.

Step 4. If $\phi(\min\{1 - x_i, x_j\}, x, i, j) \geq F(x)$, then let $\varepsilon = \min\{1 - x_i, x_j\}$, otherwise let $\varepsilon = -\min\{x_i, 1 - x_j\}$. Let $x := (x_1, \ldots, x_i + \varepsilon, \ldots, x_j - \varepsilon, \ldots, x_n)$. Goto Step 2.

Note that in Step 3, such components x_i and x_j are always found due to constraint (2). Also, in Step 4, either $\phi(\min\{1 - x_i, x_j\}, x, i, j) \geq F(x)$ or $\phi(-\min\{x_i, 1 - x_j\}, x, i, j) \geq F(x)$ holds by property (B). The above Steps 3 and 4 are called "pipage" steps. Since the number of non-binary components decreases at least by one, the pipage steps are iterated at most n times, and the obtained solution x' satisfies $F(x') \geq 3/4L(x^*)$, which implies that the algorithm is a polynomial time 3/4-approximation algorithm.

Theorem 10 k-MaxED *has a polynomial time* 3/4-*approximation algorithm if* k *is at most the size of the minimum edge dominating set.* □

References

[AS04] A.A. Ageev and M.I. Sviridenko. Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee, *Journal of Combinatorial Optimization*, **8**, pp. 307 – 328 (2004)

[AYN+05] S.I. Ao, K. Yip, M. Ng, D. Cheung, P.-Y. Fong, I. Melhado, P.C. Sham. CLUSTAG: Hierarchical Clustering and Graph Methods for Selecting Tag SNPs. *Bioinformatics*, **25**, pp. 1735 - 1736 (2005)

[AL95] S. Arora and C. Lund. Hardness of Approximation, in *Approximation Algorithms for NP-hard problems* (D.S. Hochbaum, ed), PWS Pub., pp. 399–446 (1995)

[CC06a] M. Chlebík and J. Chlebíková. Complexity of Approximating Bounded Variants of Optimization Problems. *Theoretical Computer Science*, **354**, pp. 320 – 338 (2006)

[CC06b] M. Chlebík and J. Chlebíková. Approximation Hardness of Edge Dominating Set Problems. *J. Combinatorial Optimization*, **11**, pp. 279 – 290 (2006)

[CR74] R. Church and C. ReVelle. The Maximal Covering Location Problem. *Papers of the Regional Science Association*, **32**, pp. 101 – 118 (1974)

[Fei98] U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *J. the ACM*, **45** (4), pp. 634 – 652 (1998)

[GJ79] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, New York (1979)

[GW94] M.X. Goemans and D.P. Williamson, New 3/4-approximation algorithms for MAX SAT, *SIAM J. Discrete Math.* 7, pp. 656 – 666 (1994)

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin (1988)

[HYZZ02] Q. Han, Y. Ye, H. Zhang, and J. Zhang. On Approximation of Max-Vertex-Cover. *European J. Operational Research*, **143** (2), pp. 342 – 355 (2002)

[Har69] F. Harary. *Graph Theory.* Addison-Wesley (1969)

[Hås01] J. Håstad. Some Optimal Inapproximability results. *J. ACM* **48**, pp.121–133 (2001)

[HHS98] T.W. Haynes, S.T. Hedetniemi, P.J. Slater. *Fundamentals of Domination in Graphs.* CRC (1998)

[Hoc97] D.S. Hochbaum. Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems, in *Approximation Algorithms for NP-Hard Problem* (D.S. Hochbaum, ed), PWS Pub., pp. 94 – 143 (1997)

[KMPRW02] S.O. Krumke, M.V. Marathe, D. Poensgen, S.S. Ravi, H.-C. Wirth. Budgeted Maximum Graph Coverage. In *Proc. WG2002*, pp. 321–332 (2002)

[LLP+04] M.A. Langston, L. Lin, X. Peng, N.E. Baldwin, C.T. Symons. A Combinatorial Approach to the Analysis of Differential Gene Expression Data: The Use of Graph Algorithms for

Disease Prediction and Screening. In *Methods of Microarray Data Analysis IV*, Kluwer Academic Pub., pp. 223 – 238 (2004)

[MZH83] N. Megiddo, E. Zemel, S.L. Hakimi. The Maximum Coverage Location Problem. *SIAM J. Alg. Disc. Meth.*, **4**, pp. 253 – 261 (1983)

[NSH$^+$07] C.T. Nguyen, J. Shen, M. Hou, L. Sheng, W. Miller, L. Zhang. Approximating the Spanning Star Forest Problem and Its Applications to Genomic Sequence Alignment. In *Proc. SODA2007*, pp. 645 – 654 (2007)

[PY91] C.H. Papadimitriou and M. Yannakakis. Optimization, Approximation, and Complexity Classes. *J. Computer Systems Science*, **43**, pp. 425 – 440 (1991)

[Pet94] E. Petrank. The Hardness of Approximation: Gap Location. *Computational Complexity*, **4**, pp. 133 – 157 (1994)

[YG80] M. Yannakakis and F. Gavril. Edge Dominating Sets in Graphs, *SIAM J. Appl. Math.*, **38** (3), pp. 364 – 372 (1980)

[WJ08] W.-B. Wang and T. Jiang. A New Model of Multi-Marker Correlation for Genome-Wide Tag SNP Selection. *Genome Informatics* **21**, pp. 27 – 41 (2008)