Maximum Domination Problem∗

Eiji Miyano†
Hirotaka Ono‡

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

‡ 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 the 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 \( k \)-MaxED unless \( P = NP \). We also prove that, for any constant \( \varepsilon > 0 \), there is no polynomial time \((1303/1304+\varepsilon)\)-approximation algorithm for \( k \)-MaxED unless \( P = NP \).

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, \( \bigcup_{u \in D_V} N(u) \cup D_V \) (resp., \( \bigcup_{(u,v) \in D_E} \{u,v\} \cup \{v\} \) for both \( k \)-MaxVD and \( k \)-MaxED). 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 \( k \)-MaxED (resp., \( k \)-MaxVD), 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 \( NP \)-complete, all of these optimization versions of the problems trivially turn out to be \( NP \)-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.

1.1 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 \( P = NP \). That is, the optimal bound for the approximability is obtained under \( P \neq NP \). 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).
Copyright ©2011, Australian Computer Society, Inc. This paper appeared at the 17th Computing: The Australasian Theory Symposium (CATS 2011), Perth, Australia, January 2011. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 119, Alex Potanin and Taso Viglas, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.
preserving reduction.

For \( k \)-MaxED, we show that there is also a polynomial time \((1 - \varepsilon)\)-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 \( P = NP \). The approximability can be improved into \(3/4\) in the case when \( k \) is at most the size of maximum minimal matching of the graph, which would be a natural setting for the \( k \)-MaxED problem. In the results, although the \((1 - \varepsilon)\)-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 \( NP \)-hard optimization problems, and thus there is much of literature studying their (in)approximability: Since MinVD is a special instance of \textsc{Minimum Set Cover}, it is approximable within \(1 + \log n\) for an \( n \)-vertex graph. On the other hand, unless \( P = NP \), no polynomial-time algorithm can generally provide a \((1 - \varepsilon)\log n\) approximation guarantee for any \( \varepsilon \) [Fei98]. As for MinED, there is a polynomial-time 2-approximation algorithm: One can see that any maximal matching of a graph is same size as the smallest edge dominating set [YG80]. Furthermore, the smallest maximal matching has the guarantee for any \( \ell \geq 1 + \log n \) for an \( n \)-vertex graph. On the other hand, unless \( P = NP \), no polynomial-time algorithm can generally provide a \((1 - \varepsilon)\log n\) approximation guarantee for any \( \varepsilon \) [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. 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 \( NP \)-hard to approximate MinED within \(7/6\). They also showed that it is \( NP \)-hard to approximate MinED within \((7 + \Delta(n))/6 + \Delta(n)/n\) for graphs with maximum degree at most \( \Delta \).

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, ZZ83]) 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 maximization versions have not been well studied yet: there are only few results on the (in)approximability: The \( k \)-\textsc{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 for a small positive \( \varepsilon \). (Note that, however, the specific value of \( \varepsilon \) 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 \textsc{Budgeted Maximum Graph Coverage} introduced in [KMP95] has a great resemblance to \( k \)-MaxVD and \( k \)-MaxED, but it must be quite difficult to formulate our problems in the framework of \textsc{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 \( NP \)-hard to approximate \( k \)-MaxVD within ratio \((1 - 1/e)\). Section 4 discusses the inapproximability of \( k \)-MaxED: we prove that it is \( NP \)-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 \textsc{Maximum Coverage Problem} (MaxCP), which is formalized as follows:

**MaxCP:**

**Instance:** A collection of sets \( S = \{ S_1, S_2, \ldots, S_n \} \) defined over a domain of elements \( X = \{ x_1, x_2, \ldots, x_n \} \), a weight function \( w : X \rightarrow \mathbb{Q} \) where \( \mathbb{Q} \) is the set of nonnegative rational numbers, and a positive integer \( \ell \leq |S| \).

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

For the MaxCP, a greedy heuristic is naturally considered: 

**Algorithm Greedy**

**Step 1.** Let \( S' = \emptyset \).

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

**Step 3.** Let \( S^* = \text{argmax}_{S \in S \setminus S'} \{ f(S, S') \} \), and let \( S' = S' \cup \{ S^* \} \). If \(|S'| = \ell\), output \( S' \) 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.

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 \( S_i = \{ S_{w_i} \} \cup \{ v_i \} \) 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 \( S_i = \{ S_{w_i} \} \cup \{ u_i \} \) if \( i \in N(u) \cup \{ v, i \} \) or \( i \in N(v) \cup \{ u, i \} \) or \( w(u, v) = 1 \) 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 the constant approximation ratio better than \((1 - 1/e)\) exists for \(\text{MaxCP}\) unless \(\mathcal{P} = \mathcal{NP}\), even if the weights of elements are restricted to 1. □

By Theorems 1 and 2, the above simple greedy heuristic algorithm for \(\text{MaxCP}\) 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 \(k\)-\text{MaxVD}\) and \(k\)-\text{MaxED}.

3 Inapproximability of \(k\)-\text{MaxVD}

In this section, we discuss the hardness of approximation of \(k\)-\text{MaxVD}. As seen above, an instance of \(k\)-\text{MaxVD} is easily formulated as an instance of \(\text{MaxCP}\), 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 \(k\)-\text{MaxVD}.

However, we can still show that \(k\)-\text{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 \(\text{MaxCP}\) whose weight is uniform (\(\text{MaxUCP}\), for short), which also has the inapproximability bound \((1 - 1/e)\).

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

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

1. if \(OPT_{\text{mcp}}(I) = \max\), then \(OPT_{\text{mvd}}(G) \geq \ell(q + 1) + p \cdot \max\), and
2. if \(OPT_{\text{mcp}}(I) \leq (1 - 1/\epsilon) \max\), then \(OPT_{\text{mvd}}(G) \leq (1 - 1/\epsilon + \ell/(q + 1 + p \cdot \max))((q + 1) + p \cdot \max)\), where \(p, q\) are any positive integer larger than \(1\).

Proof: Consider an instance \(I\) of \(\text{MaxUCP}\); a collection of sets \(S = \{S_1, S_2, \ldots, S_m\}\) defined over a domain of elements \(X = \{x_1, x_2, \ldots, x_n\}\), and a positive integer \(q\). We construct the following graph, illustrated in Figure 1. Let \(V_G = \{r_{1,1}, r_{1,2}, \ldots, r_{n,q}, r_{1,2}, r_{1,2}, r_{n,q}, 1, \ldots, m\}\) be a set of \(nq\) nodes associated with \(|S| = 1\). Also, let \(V_S = \{\alpha_{1,2}, \alpha_{2,3}, \ldots, \alpha_{m}\}\) be a set of \(m\) nodes corresponding to the \(m\) sets, \(S_i\) through \(S_m\), and let \(V_X = \{\beta_{1}, \beta_{1,2}, \ldots, \beta_{1,2}, \ldots, \beta_{1,2}, \ldots, \beta_{1,2}\}\) be a set of \(np\) nodes, in which \(\beta_{1,2}, \ldots, \beta_{1,2}\) correspond to \(x_i\). These \(V_F, V_S,\) and \(V_X\) are pairwise disjoint, and let \(V_G = V_F \cup V_S \cup V_X\). The set \(E\) of edges is defined as follows:

- \(E_1 = \{\alpha_{i,1}, \alpha_{1,2}, \ldots, \alpha_{m}\}\)
- \(E_2 = \{\beta_{1,1}, \beta_{1,2}, \ldots, \beta_{1,2}\} \times \{x_i\}\)

Finally, we set \(k = \ell\). This reduction can be done in polynomial time if both \(k\) and \(q\) are bounded by some polynomial of \(|I|\).

(i) Suppose that \(OPT_{\text{mcp}}(I) = \max\) and the optimal collection of sets is \(OPT = \{S_{i_1}, S_{i_2}, \ldots, S_{i_t}\}\) where \(\{i_1, i_2, \ldots, i_t\} \subseteq \{1, 2, \ldots, m\}\). Then, if we select a set \(D = \{\alpha_{i_1,1}, \alpha_{i_2,1}, \ldots, \alpha_{i_t,1}\}\) of \(k = \ell\) nodes as a partial dominating set, then we can obtain the dominated set with size \((q + 1) + p \cdot \max\); since the nodes \(\alpha_{i,j}\)’s are connected with exactly \(p \cdot \max\) edges and each \(\alpha_{i,j}\) is connected with exactly \(q\) \(r\)’s.

(ii) We show this by contradiction. We first suppose that \(OPT_{\text{mcp}}(I) \leq (1 - 1/\epsilon) \max\) but \(OPT_{\text{mvd}}(G) > (1 - 1/\epsilon + \ell/(q + 1 + p \cdot \max))((q + 1) + p \cdot \max)\) 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 in \(D_X\) can dominate itself and vertices in \(\{x_i\}\) satisfying \(x_i \in S_i\), 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 \(2|D_R| + (m + 1)|D_X| + (q + 1)|D_S|\) vertices, where \(|D_R| + |D_X| + |D_S| = \ell\). It follows that \(ps + 2|D_R| + (m + 1)|D_X| + (q + 1)|D_S|) \geq OPT_{\text{mvd}}(G) > (1 - 1/\epsilon) \max + \ell(q + 1)\). Since \(q + 1) = ((|D_R| + |D_X| + |D_S|)(q + 1) > 2|D_R| + (m + 1)|D_X| + (q + 1)|D_S|), we have \(s > (1 - 1/\epsilon) \cdot \max\), which contradicts \(OPT_{\text{mcp}}(I) \leq (1 - 1/\epsilon) \cdot \max\). □

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 \((1 - 1/\epsilon)\) for \(k\)-\text{MaxVD} unless \(\mathcal{P} = \mathcal{NP}\). □

4 Inapproximability of \(k\)-\text{MaxED}

4.1 Degree Bounded Graphs

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

\(k\)-\text{MaxVC}(\Delta):

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

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

Theorem 5 ([Pet94]) \(k\)-\text{MaxVC}(\Delta) is \(APX\)-hard for \(\Delta \geq 3\), that is, there exists a constant \(\varepsilon_\Delta\) such that \(k\)-\text{MaxVC}(\Delta) \text{ has no polynomial time approximation algorithm whose approximation factor is better than } 1 - \varepsilon_\Delta \text{ unless } \mathcal{P} = \mathcal{NP}\). □

It is important to note that the hardness parameter \(\varepsilon_\Delta\) of approximation comes from the gap-preserving identity reduction from Minimum Vertex Cover with Bound Degree \(\langle \text{MinVC}(\Delta)\rangle\), and the hardness result of approximation for \(\text{MinVC}(\Delta)\) shown in [PY91]. The reduction implies that if an algorithm \(ALG\) for \(k\)-\text{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
with at most \( k + \varepsilon \Delta |E| \) vertices for MinVC. Thus, the approximation ratio of ALG for MinVC is obtained as follows:

\[
\frac{\text{ALG}}{\text{OPT}} \leq 1 + \varepsilon \frac{\Delta |E|}{k} \leq 1 + \varepsilon \frac{\Delta |V| \Delta}{2k},
\]

where OPT denotes an optimal algorithm for MinVC. The second inequality comes from \(|E| \leq |V| \cdot \Delta / 2\). 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:

\[
\varepsilon_a \geq \frac{2}{297} \cdot \frac{k}{|V|} \quad \text{and} \quad k \geq \frac{958}{1048} \cdot |V|.
\]

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

\[
1 - \varepsilon_a \leq \frac{309340}{311256} \leq 0.993845 \leq \frac{162}{163}.
\]

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

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

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

(i) Suppose that OPT\( _{vc} (G) = \max \) and the optimal subset of vertices is \( V^* = \{v_1,v_2,\ldots,v_r\} \) where \( \{1^*,2^*,\ldots,\ell^*\} \subseteq \{1,2,\ldots,n\} \). Then, we select an edge set \( D = \{(v_1,v_1'),(v_2,v_2'),\ldots,(v_r,v_r')\} \) of \( \ell \) edges as a partial edge dominating set. One can see that \( D \) dominates \( (2\Delta + 1) + \max \) since the nodes \( v_i \)'s are connected with exactly \( \max \) edges in \( E \) and each \( v_i' \) is connected with exactly \( 2\Delta \) u's. Therefore, \( \ell(2\Delta + 1) + \max \) holds. Let \( D \) be an optimal partial edge dominating set of \( G \). We partition \( D \) into \( D_{org} = D \cap E \) and \( D_{org} = D \cap E \). We then estimate the number of dominated edges. We can see that an edge \( (v_i,v_j) \) in \( D \) dominates itself, the other edges \( (v_i',v_j') \) 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 \( D \) can dominate at most \( (2\Delta + 1)|D| \). As for \( D_{org} \), it can be considered a subset of \( V \), which covers some edges in \( E \). Let \( s \) denote the number of the covered edges by \( D_{org} \). Then \( D_{org} \) can dominate at most \( (2\Delta + 1)|D| + s \) edges, because each edge \( (v_i,v_j) \) in \( E \) dominates itself and its adjacent \( \ell \) edges. Therefore, the following holds that

\[
\ell(2\Delta + 1) + s \geq c + 2\Delta + 1 \quad \text{and} \quad \ell(2\Delta + 1) + c \cdot \max.
\]

(ii) Suppose that OPT\( _{vc} (G) \leq c \cdot \max \) but OPT\( _{med} (G') > \frac{c + 2\Delta + 1}{2\Delta + 2} (\ell(2\Delta + 1) + \max) \) holds. Let \( D \) be an optimal partial edge dominating set of \( G \). We partition \( D \) into \( D_{org} = D \cap E \) and \( D_{org} = D \cap E \). We then estimate the number of dominated edges. We can see that an edge \( (v_i,v_j) \) in \( D \) dominates itself, the other edges \( (v_i',v_j') \) 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 \( D \) can dominate at most \( (2\Delta + 1)|D| + s \) edges, because each edge \( (v_i,v_j) \) in \( E \) dominates itself and its adjacent \( \ell \) edges. Therefore, the following holds that

\[
\ell(2\Delta + 1) + s \geq c + 2\Delta + 1 \quad \text{and} \quad \ell(2\Delta + 1) + c \cdot \max.
\]

(iii) Suppose that OPT\( _{vc} (G) = \max \) and the optimal subset of vertices is \( V^* = \{v_1,v_2,\ldots,v_r\} \) where \( \{1^*,2^*,\ldots,\ell^*\} \subseteq \{1,2,\ldots,n\} \). Then, we select an edge set \( D = \{(v_1,v_1'),(v_2,v_2'),\ldots,(v_r,v_r')\} \) of \( \ell \) edges as a partial edge dominating set. One can see that \( D \) dominates \( (2\Delta + 1) + \max \) since the nodes \( v_i \)'s are connected with exactly \( \max \) edges in \( E \) and each \( v_i' \) is connected with exactly \( 2\Delta \) u's.

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(\( \Delta \)), that is, there is no polynomial time approximation algorithm with any constant factor better than \( c(\Delta) \) unless \( P = NP \). 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 \( P = NP \).
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 the 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 $L(x^*)$. Then we 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:

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

where $F(x) = \sum_{e \in \{(u, v) \in E | x_e \neq 1\}} w_e (1 - \prod_{i \in N(u)} (1 - x_{(u,i)}) \prod_{i \in N(v)} (1 - x_{(v,i)}))$. By considering $x_e = 1$ if $e \in D_E$ and $x_e = 0$ otherwise, it is easy to see that the optimal solution 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

$$L(x) = \sum_{e \in \{(u, v) \in E | x_e \neq 1\}} w_e \times \min(1, \sum_{i \in N(u)} x_{(u,i)} + \sum_{i \in N(v)} (x_{(v,i)} - x_{(u,i)})).$$

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:

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

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 $\varepsilon$ in polynomial time, for a given any $\varepsilon > 0$ ($\varepsilon$ is a part of the input, so the running time dependence on $\varepsilon$ is polynomial in $\log 1/\varepsilon$). This can be done through the ellipsoid algorithm [GLSS88]. Here is an SDP-formulation of the problem (SDP).

$$\begin{align*}
\text{max} & \quad L'(x) \overset{\text{def}}{=} \sum_{e \in E} w_e z_e \cdot y_0 + \frac{1}{2} \\
\text{subject to} & \quad \sum_{e \in E} y_e \cdot y_0 = 2k - m, \\
& \quad \sum_{e \in (v,i) \in \Gamma(v)} y_e \cdot y_0 \\
& \quad \forall e \in (i,j) \in E \quad (7) \\
& \quad y_e, x_e \in B, \quad \forall \, e \in E \\
& \quad y_0 \in B, \\
\end{align*}$$

(5)

where $B$ is the unit sphere in $\mathbb{R}^{2m-1}$, that is, $B = \{y \in \mathbb{R}^{2m-1} \mid \|y\|_2 = 1\}$. Namely, $y_e$‘s and $x_e$‘s are unit vectors in the $\mathbb{R}^{2m-1}$ 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 $L(x)$. 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

$$x_e = \frac{1 - y_e \cdot y_0}{2}.$$
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

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

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

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

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

It is easy to see that property (B) holds, because $\phi(\epsilon, x, i, j)$ is a quadratic polynomial in $\epsilon$, 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 $\epsilon = \min\{1 - x_i, x_j\}$, otherwise let $\epsilon = -\min\{x_i, 1 - x_j\}$. Let $x := (x_1, \ldots, x_i + \epsilon, \ldots, x_j - \epsilon, \ldots, x_n)$. 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. \(\square\)

**References**


Disease Prediction and Screening. In Methods of 
Microarray Data Analysis IV, Kluwer Academic 

[MZH83] N. Megiddo, E. Zemel, S.L. Hakimi. The 
Maximum Coverage Location Problem. SIAM J. 

W. Miller, L. Zhang. Approximating the Span-
ning Star Forest Problem and Its Applica-
tions to Genomic Sequence Alignment. In Proc. 

Optimization, Approximation, and Complexity 

[Pet94] E. Petrank. The Hardness of Approximation: 
133 – 157 (1994)

[YG80] M. Yannakakis and F. Gavril. Edge Dominat-
ing Sets in Graphs. SIAM J. Appl. Math., 38 (3), 

[WJ08] W.-B. Wang and T. Jiang. A New Model of 
Multi-Marker Correlation for Genome-Wide Tag 