Information Processing Letters 79 (2001) 261–266

Information Processing Letters

www.elsevier.com/locate/ipl

On approximability of the independent/connected edge dominating set problems

Toshihiro Fujito

Department of Electronics, Nagoya University, Furo, Chikusa, Nagoya, 464-8603 Japan

Received 16 June 2000; received in revised form 5 December 2000
Communicated by K. Iwama

Abstract

We investigate polynomial-time approximability of the problems related to edge dominating sets of graphs. When edges are unit-weighted, the edge dominating set problem is polynomially equivalent to the minimum maximal matching problem, in either exact or approximate computation, and the former problem was recently found to be approximable within a factor of 2 even with arbitrary weights. It will be shown, in contrast with this, that the minimum weight maximal matching problem cannot be approximated within any polynomially computable factor unless P = NP.

The connected edge dominating set problem and the connected vertex cover problem also have the same approximability when edges/vertices are unit-weighted. The former problem is already known to be approximable, even with general edge weights, within a factor of 3.55. We will show that, when general weights are allowed, (1) the connected edge dominating set problem can be approximated within a factor of $3 + \varepsilon$, and (2) the connected vertex cover problem is approximable within a factor of $\ln n + 3$ but cannot be within $(1 - \varepsilon) \ln n$ for any $\varepsilon > 0$ unless NP $\subset$ DTIME$(n^{O(\log \log n)})$.

Keywords: Combinatorial optimization problems; Approximation algorithms; Edge dominating set

1. Introduction

In this paper we investigate polynomial-time approximability of problems related to edge dominating sets of graphs. For two pairs of problems considered, it will be shown that, while both problems in each pair have the same approximability for the unweighted case, their approximation properties differ drastically when general non-negative weights are allowed.

In an undirected graph an edge *dominates* all the edges adjacent to it, and an *edge dominating set* (eds) is a set of edges collectively dominating all the other edges in a graph. The problem EDS is then that of finding a smallest eds or, if edges are weighted, an eds of minimum total weight. Yannakakis and Gavril showed that EDS is NP-complete even when the graphs are planar or bipartite of maximum degree 3 [20]. A set of edges is called a *matching* (or *independent*) if no two of them have a vertex in common, and a matching is *maximal* if no other matching properly contains it. Notice that any maximal matching is necessarily an eds, because an edge not in it must be adjacent to some in it. For this reason it is also called an *independent edge dominating set*, and the problem IEDS asks for a minimum maximal matching in a given graph. Interestingly, one can also construct a

E-mail address: fujito@nuee.nagoya-u.ac.jp (T. Fujito).

0020-0190/01/$ – see front matter
PII: S0020-0190(01)00138-7

maximal matching, from any eds, of no larger size in polynomial time [11], implying that EDS and IEDS are polynomially equivalent, in exact or approximate computation, when graphs are *unweighted*. Based on this and the fact that any maximal matching cannot be more than twice larger than any maximal matching, it has been long known that either problem, without weights, can be approximated within a factor of 2. Moreover, weighted EDS was recently shown approximable within a factor of 2 [5,3]. In contrast with this, we will present strong inapproximability results for weighted IEDS.

We next consider EDS with connectivity requirement, called the *connected edge dominating set* (CEDS) problem, where it is asked to compute a connected eds (ceds) of minimum weight in a given connected graph. Since it is always redundant to form a cycle in a ceds, the problem can be restated as that of going after a minimum tree whose vertices "cover" all the edges in a graph, and thus, it is also called *tree cover*. The *vertex cover* (VC) problem is another basic NP-complete graph problem [13], in which a minimum vertex set is sought in G s.t. every edge of G is incident to some vertex in the set. When a vertex cover is additionally required to induce a connected subgraph in a given connected graph, the problem is called *connected vertex cover* (CVC) and known to be as hard to approximate as VC is [6].

These problems are closely related to EDS and CEDS in that an edge set F is an eds for G iff $V(F)$, the set of vertices touched by edges of F, is a vertex cover for G, and similarly, a tree F is a ceds iff $V(F)$ is a connected vertex cover. It follows that since the number of vertices is that of edges plus one in any tree, the unweighted versions of CEDS and CVC have the same approximability, and in fact they are known to be approximable within a factor of 2 [18,1]. It is not known, however, if CEDS and CVC can be somehow related even if general weights are allowed. An algorithm scheme of Arkin et al. for weighted CEDS gives its approximation factor in the form of $r_{\mathrm{St}} + r_{\mathrm{wvc}}(1 + 1/k)$, for any constant k, where r_{St} (r_{wvc}) is the performance ratio of any polynomial time algorithm for the Steiner tree (weighted vertex cover, respectively) problem [1]. By using the currently best algorithms for Steiner tree [17] and for weighted vertex cover [2] in their scheme, the bound for weighted CEDS is estimated at 3.55. After improving this bound to $3 + \varepsilon$, we will show that weighted CVC is as hard to approximate as weighted set cover, indicating that it is not approximable within a factor better than $(1 - \varepsilon)\ln n$ unless

$$\mathrm{NP} \subset \mathrm{DTIME}\big(n^{\mathrm{O}(\log\log n)}\big)$$

[4]. Lastly, we present an algorithm approximating weighted CVC within a factor of

$$H(\Delta - 1) + r_{\mathrm{wvc}} \leqslant \ln(\Delta - 1) + 3,$$

where $H(k)$ is the kth harmonic number and Δ is the maximal vertex degree of a graph. A related problem, *connected dominating set*, can be approximated within a factor of $\ln n + 3$ for the unweighted case [8], but when weighted, the best known bound is $1.35 \ln n$ [9].

2. Independent edge dominating set

To show that it is extremely hard to approximate IEDS, we use a reduction from 3SAT. Let us first describe a general construction of graph G_ϕ for a given 3SAT instance (i.e., a CNF formula) ϕ, by adapting the one used in reducing SAT to minimum maximal independent set [12,10] to our case. For simplicity, every clause of ϕ is assumed without loss of generality to contain exactly three literals. Each variable x_i appearing in ϕ is represented in G_ϕ by two edges adjacent to each other, and the endvertices of such a path of length 2 are labeled x_i and $\bar{x}_i$; let E_v denote the set of these edges. Each clause c_j of ϕ is represented by a triangle (a cycle of length 3) C_j, and vertices of C_j are labeled distinctively by literals appearing in c_j; let E_c denote the set of edges in these disjoint triangles. The paths in E_v and triangles in E_c are connected together by having an edge between every vertex of each triangle and the endvertex of a path having the same label. The set of these edges lying between E_v and E_c is denoted by E_b. It is a simple matter to verify that, for a 3SAT instance ϕ with m variables and p clauses, G_ϕ constructed this way consists of $3(m + p)$ vertices and $2(m + 3p)$ edges.

Lemma 1. *Let $M(G)$ denote a minimum maximal matching M in G. For any* 3SAT *instance ϕ with m variables and p clauses, and for any number t, there exists a graph G_ϕ on $3(m+p)$ vertices and $2(m+3p)$*

edges, and a weight assignment $w: E \to \{1, t\}$*, such that*

$$w\big(M(G_\phi)\big) \begin{cases} \leqslant m + p, & \text{if } \phi \text{ is satisfiable,} \\ > t, & \text{otherwise.} \end{cases}$$

Proof. Let $w(e) = 1$ if $e \in E_v \cup E_c$ and $w(e) = t$ if $e \in E_b$. Suppose that ϕ is satisfiable, and let τ be a particular truth assignment satisfying ϕ. Construct a matching M_τ in E_v by choosing, for each i, the edge with its endvertex labeled by x_i if $\tau(x_i)$ is true and the one having an endvertex labeled by $\bar{x}_i$ if $\tau(x_i)$ is false. Consider any triangle C_j in E_c. Since τ satisfies ϕ, at least one edge among those in E_b connecting C_j and E_v must be dominated by M_τ. This means that all the edges in E_b between C_j and E_v can be dominated by M_τ, plus one edge on C_j. Let M_c denote the set of such edges, each of which taken this way from each C_j. Then, $M_\tau \cup M_c$ is clearly a minimal matching since it dominates all the edges in G_ϕ. Since all the edges in $M_\tau \cup M_c$ are of weight 1, its weight is $|M_\tau \cup M_c| = m + p$. On the other hand, if ϕ is not satisfiable, there is no way to dominate all the edges in E_b only by any matching built inside $E_v \cup E_c$, and hence, any maximal matching in G_ϕ must incur a cost of more than t. □

The computational hardness of approximating weighted IEDS easily follows from this lemma:

Theorem 2. *For any polynomial time computable function* $\alpha(n)$*, IEDS cannot be approximated on graphs with* n *vertices within a factor of* $\alpha(n)$*, unless* P = NP.

Proof. Given a 3SAT instance ϕ with m variables and p clauses, construct a graph G_ϕ and assign a weight $w(e) \in \{1, t\}$ to each edge $e \in E$, as in the proof of Lemma 1. Since G_ϕ consists of $3(m + p)$ vertices and $(m + p)\alpha(3(m + p))$ is computable in time polynomial in the length of ϕ, $m + 3p$, we can set $t = (m + p)\alpha(3(m + p)) = (m + p)\alpha(n)$. If a polynomial time algorithm A exists approximating IEDS within a factor of $\alpha(n)$, then, when applied to G_ϕ, A will output a number at most $(m + p)\alpha(n)$ if ϕ is satisfiable, and a number greater than $t = (m + p)\alpha(n)$ if ϕ is not satisfiable. Hence, A decides 3SAT in polynomial time. □

3. Connected edge dominating set

We first consider a restricted version of CEDS; for a designated vertex r called *root*, an r-ceds is a ceds touching r, and the problem r-CEDS is to compute an r-ceds of minimum weight. Given an undirected graph $G = (V, E)$ with edge weights $w: E \to \mathbb{Q}_+$, let $\vec{G} = (V, \vec{E})$ denote its directed version obtained by replacing each edge $\{u, v\}$ of G by two directed ones, (u, v) and (v, u), each of weight $w(\{u, v\})$. For the root r, a non-empty set $S \subseteq V - \{r\}$ is called *dependent* if S is not an independent set in G. Suppose $T \subseteq E$ is an r-ceds, and let $\vec{T}$ denote the directed counterpart obtained by choosing, for each pair of directed edges, the one directed away from the root to a leaf. Clearly, $w(T) = w(\vec{T})$. Moreover, let $\vec{T}$ be represented by its characteristic vector $x^{\vec{T}} \in \{0, 1\}^{\vec{E}}$, and, for any $x \in \mathbb{Q}^{\vec{E}}$ and $\vec{F} \subseteq \vec{E}$, let $x(\vec{F}) \sum_{a \in \vec{F}} x_a$. Then, $x^{\vec{T}}$ satisfies the linear inequality $x(\delta^-(S)) \geqslant 1$ for all dependent sets $S \subseteq V - \{r\}$, where $\delta^-(S) = \{(u, v) \in \vec{E} \mid u \notin S, v \in S\}$, because, when an edge exists inside S, at least one arc of $\vec{T}$ must enter it. Thus, the following linear programming problem is a relaxation of r-CEDS:

$$\begin{aligned} & Z_{\text{ceds}} = \min \textstyle\sum_{a \in \vec{E}} w(a) x_a \\ & \text{subject to:} \\ & x\big(\delta^-(S)\big) \geqslant 1 \quad \forall \text{ dependent set } S \subseteq V - \{r\}, \\ & 0 \leqslant x_a \leqslant 1 \qquad \forall a \in \vec{E}. \end{aligned} \tag{1}$$

Lemma 3. *For any feasible solution* $x \in \mathbb{Q}^{\vec{E}}$ *of* (1)*, let* $V_+(x) = \{u \in V \mid x(\delta^-(\{u\})) \geqslant 1/2\}$*. Then,* $V_+(x) \cup \{r\}$ *is a vertex cover for* G.

Proof. Take any edge $e = \{u, v\} \in E$ with $r \notin e$. Then, $\{u, v\}$ is a dependent set, and $x(\delta^-(\{u, v\})) \geqslant 1$, which implies either $x(\delta^-(\{u\}))$ or $x(\delta^-(\{v\}))$ is at least $1/2$. Thus, $\{u, v\} \cap V_+(x) \neq \emptyset$. □

From this lemma it is clear that any tree $T \subseteq E$ containing all the vertices in $V_+(x) \cup \{r\}$ is an r-ceds for G, and, in searching for such T of small weight, it can be assumed without loss of generality that the edge weights satisfy the triangle inequality since any edge between two vertices can be replaced, if necessary, by the shortest path between them. Then, the problem of finding such a tree of minimum weight is called the (*metric*) *Steiner tree* problem: Given $G=(V,E)$ with edge weight $w: E \to \mathbb{Q}_+$ and a set $R \subseteq V$ of *required vertices* (or *terminals*), find a minimum weight tree containing all the required vertices and any others (called *Steiner vertices*). For this problem Rajagopalan and Vazirani considered the so-called *bidirected cut relaxation* [16]:

$$
\begin{aligned}
&Z_{\mathrm{smt}} = \min \textstyle\sum_{a \in \vec{E}} w(a) x_a \\
&\text{subject to:} \\
&x\big(\delta^-(S)\big) \geqslant 1 \quad \forall \text{ valid set } S \subseteq V - \{r\}, \\
&0 \leqslant x_a \leqslant 1 \qquad \forall a \in \vec{E},
\end{aligned} \tag{2}
$$

where the root r is any required vertex and a set $S \subseteq V - \{r\}$ is *valid* if it contains a required vertex. Based on this relaxation, they designed a primal-dual approximation algorithm for metric Steiner tree and showed that it computes a Steiner tree of cost at most $(3/2+\varepsilon)Z_{\mathrm{smt}}$ and that the integrality gap of (2) is at most 3/2, when restricted to graphs in which Steiner vertices form independent sets (called *quasi-bipartite* graphs). Our algorithm for r-CEDS is now described as:

1. Compute an optimal solution x for (1).
2. Let $V_+(x) = \{u \in V \mid x(\delta^-(\{u\})) \geqslant 1/2\}$.
3. Compute a Steiner tree T with $R = V_+(x) \cup \{r\}$, as the set of required vertices, by the algorithm of Rajagopalan and Vazirani.
4. Output T.

It is clear that this algorithm computes an r-ceds for G, except for one special case in which $R = \{r\}$ and so, $T = \emptyset$; but in that case it is trivial to find an optimal r-ceds since G is a *star* centered at r. Not so clear from this description is polynomiality of its time complexity, and more specifically, that of step 1. It can be polynomially implemented by applying the ellipsoid method to (1), if the separation problem for the polytope P_{ceds} corresponding to the feasible region of (1), is solved in polynomial time [7]. So, let y be a vector in $\mathbb{Q}^{\vec{E}}$. It is easily tested if $0 \leqslant y_a \leqslant 1$ for all $a \in \vec{E}$. To test whether $y(\delta^-(S)) \geqslant 1$ for every dependent set S, we consider y as a capacity function on the arcs of $\vec{G}$. For every arc a, not incident upon r, contract a by merging its two endvertices into a single vertex v_a, and determine an (r, v_a)-cut C_a of minimum capacity. It is then rather straightforward to see that

$$
\begin{aligned}
&\min\big\{y(C_a) \mid a \in \vec{E} - \delta(\{r\})\big\} \\
&\quad = \min\big\{y(\delta^-(S)) \mid S \subseteq V - \{r\} \text{ is dependent}\big\},
\end{aligned}
$$

where $\delta(\{r\})$ is the set of arcs incident to r. So, by calculating $|E - \delta(\{r\})|$ minimum capacity (r, v_a)-cuts, we can find a dependent set S of minimum cut capacity $y(\delta^-(S))$. If $y(\delta^-(S)) \geqslant 1$, we conclude that $y \in P_{\mathrm{ceds}}$, while, if not, the inequality $x(\delta^-(S)) \geqslant 1$ is violated by y and a separation hyperplane is found.

Notice that our graph G is quasi-bipartite when $V_+(x) \cup \{r\}$ is taken as the set of required vertices since it is a vertex cover for G, and for the approximation quality of solutions, we have [1]

Theorem 4. *The algorithm above computes an r-ceds of weight at most* $(3+\varepsilon)Z_{\mathrm{ceds}}$.

Proof. Let $x \in \mathbb{Q}^{\vec{E}}$ be an optimal solution of (1), and T be an r-ceds computed by the algorithm. As mentioned above, it was shown that $w(T) \leqslant (3/2+\varepsilon)Z_{\mathrm{smt}}$ when graphs are quasi-bipartite [16]. So, it suffices to show that $2x$ is a feasible solution of (2) with $R = V_+(x) \cup \{r\}$ for then,

$$
Z_{\mathrm{smt}} \leqslant 2 \sum_{a \in \vec{E}} w(a) x_a = 2 Z_{\mathrm{ceds}},
$$

and hence,

$$
w(T) \leqslant 2(3/2+\varepsilon) Z_{\mathrm{ceds}}.
$$

To this end, let $S \subseteq V - \{r\}$ be any valid set. If S is not an independent set in G, it is dependent, ensuring that $x(\delta^-(S)) \geqslant 1$. Suppose now S is an independent set. Since it is a valid set, S contains a vertex $u (\neq r)$ in $V_+(x)$. But then, $x(\delta^-(\{u\})) \geqslant 1/2$, and, since S is an independent set in G, $2x(\delta^-(S)) \geqslant 2x(\delta^-(\{u\})) \geqslant 1$. Thus, in either case, $2x$ satisfies all the linear constraints of (2). □

Since the integrality gap of (2) is bounded by 3/2 for quasi-bipartite graphs, we have

Corollary 5. *The integrality gap of* (1) *is at most* 3.

[1] Independently of our work, Koenemann et al. recently obtained the same performance guarantee by essentially the same algorithm [15].

Lastly, since any ceds is an r-ceds for some $r \in V$, by applying the algorithm with $r = u$ for each $u \in V$ and taking the best one among all computed, CEDS can be approximated within a factor of $3 + \varepsilon$.

4. Connected vertex cover

Savage showed that non-leaf vertices of any depth first search tree form a vertex cover of size at most twice the smallest size [18]. Since such a vertex cover clearly induces a connected subgraph, it actually means that a cvc of size no more than twice larger than the smallest vertex cover always exists and can be efficiently computed. When vertices are arbitrarily weighted, however, the weighted set cover problem can be reduced to it in an approximation preserving manner, as was done for node-weighted Steiner trees [14] and connected dominating sets [8]:

Theorem 6. *The weighted set cover problem can be approximated within the same factor as the one within which weighted CVC can be on bipartite graphs.*

Proof. From a set cover instance $(U, \mathcal{F})$ and $w : \mathcal{F} \to \mathbb{Q}_+$, where $\mathcal{F} \subseteq 2^U$ and $\bigcup_{S \in \mathcal{F}} S = U$, construct a bipartite graph G as a CVC instance, using a new vertex c, with vertex set $(U \cup \{c\}) \cup \mathcal{F}$ s.t. an edge exists between c and every $S \in \mathcal{F}$, and between $u \in U$ and $S \in \mathcal{F}$ iff $u \in S$. All the vertices in U and c are assigned zero weights, while every vertex $S \in \mathcal{F}$ inherits $w(S)$, the weight of set S, from $(U, \mathcal{F})$.

For a vertex subset V' of G let $\Gamma(V')$ denote the set of vertices adjacent to a vertex in V'. Clearly, $\mathcal{F}' \subseteq \mathcal{F}$ is a set cover for $(U, \mathcal{F})$ iff $U \subseteq \Gamma(\mathcal{F}')$ in G, and moreover, for any set cover $\mathcal{F}$, $\mathcal{F}' \cup U \cup \{c\}$ is a cvc of the same weight. On the other hand, for any cvc C for G, $U \subseteq \Gamma(C \cap \mathcal{F})$, i.e., $C \cap \mathcal{F}$ is a set cover, of the same weight because, if $u \notin \Gamma(C \cap \mathcal{F})$ for some $u \in U$, $\Gamma(\{u\}) \cap C = \emptyset$, and hence, there is no way to properly cover an edge incident to u by C. Thus, since it costs nothing to include c and vertices in U, any cvc for G can be assumed to be of the form $\mathcal{F}' \cup U \cup \{c\}$ s.t. $\mathcal{F}'$ is a set cover for $(U, \mathcal{F})$, with its weight equaling to that of $\mathcal{F}'$. Therefore, any algorithm approximating CVC within a factor r can be used to compute a set cover of weight at most r times the optimal weight. □

Due to the non-approximability of set cover [4], it follows that

Corollary 7. *The weighted CVC cannot be approximated within a factor better than* $(1 - \varepsilon)\ln n$ *for any* $\varepsilon > 0$, *unless* $\mathrm{NP} \subset \mathrm{DTIME}(n^{\mathrm{O}(\log\log n)})$.

One simple strategy for approximating weighted CVC, which turns out to yield a nearly tight bound, is to compute first a vertex cover $C \subseteq V$ for $G = (V, E)$, and then to augment it to become connected by an additional vertex set $D \subseteq V - C$. While many good approximation algorithms are known for vertex cover, we also need to find such D of small weight. Although this problem is not exactly same as weighted set cover, it can be seen as a specialization of the *submodular set cover* problem: Given a finite set N and a nondecreasing, submodular set function $f : 2^N \to \mathbb{R}_+$, compute $D \subseteq N$ of minimum weight such that $f(D) = f(N)$. For our case, consider the complete graph H with its vertex set consisting of all the components in $G[C]$, and for each $u \in V - C$, let E_u be any edge set of H spanning all and only the neighboring components of u. Take $N = V - C$ and define f on 2^N such that $f(D) = r(\bigcup_{u \in D} E_u)$, where r is the rank function of the circuit matroid of H. Such a construction ensures f be nondecreasing and submodular. Moreover, using the fact that $V - C$ is an independent set in G, it can be shown that $f(D) = \kappa(C) - \kappa(C \cup D)$, where $\kappa(F)$ denotes the number of connected components in the subgraph $G[F]$ induced by F. Since $G[C \cup D]$ is connected iff $f(D) = \kappa(C) - 1 = f(V - C)$, the problem of computing minimum $D \subseteq V - C$ such that $G[C \cup D]$ is connected, is formulated exactly by the submodular set cover problem for $(V - C, f)$. The greedy algorithm for this case is then described as:

1. Initialize $D \leftarrow \emptyset$.
2. While $G[C \cup D]$ is not connected do
3. Let u be a vertex maximizing $(\kappa(C \cup D) - \kappa(C \cup D \cup \{v\}))/w(v)$ among $v \in V - C$.
4. Set $D \leftarrow D \cup \{u\}$.
5. Output D.

It was shown by Wolsey that the performance of the greedy algorithm for submodular set cover generalizes the one for set cover:

Theorem 8 [19]. *The greedy algorithm for submodular set cover computes a solution of weight bounded by $H(\max_{j\in N} f(\{j\}))$ times the minimum weight.*

Since $\max_{j\in N} f(\{j\}) \leqslant \Delta - 1$, in our case, for a graph of maximal vertex degree Δ, the greedy heuristic works with an approximation factor bounded by $H(\Delta-1) \leqslant 1+\ln(\Delta-1)$.

Theorem 9. *The algorithm above computes a cvc of weight at most $r_{\mathrm{wvc}} + H(\Delta-1) \leqslant \ln(\Delta-1)+3$ times the minimum weight.*

Proof. Let C^* be an optimal cvc and $C \cup D$ be the one computed by the algorithm above for G, where C is a vertex cover of weight at most r_{wvc} times that of the minimum vertex cover, and D is the greedy submodular set cover for $(V - C, f)$. Clearly, $w(C) \leqslant r_{\mathrm{wvc}} w(C^*)$. Observe that $G[C^* \cup C]$ remains connected because any superset of a cvc is still a cvc. But then, it means that $C^* - C \subseteq V - C$ is a submodular set cover for $(V - C, f)$. We thus conclude that

$$\begin{aligned} w(C \cup D) &\leqslant r_{\mathrm{wvc}} w(C^*) + H(\Delta-1) w(C^* - C) \\ &\leqslant \bigl(r_{\mathrm{wvc}} + H(\Delta-1)\bigr) w(C^*). \qquad \square \end{aligned}$$

References

[1] E.M. Arkin, M.M. Halldórsson, R. Hassin, Approximating the tree and tour covers of a graph, Inform. Process. Lett. 47 (1993) 275–282.

[2] R. Bar-Yehuda, S. Even, A local-ratio theorem for approximating the weighted vertex cover problem, in: Annals of Discrete Mathematics, Vol. 25, North-Holland, Amsterdam, 1985, pp. 27–46.

[3] R. Carr, T. Fujito, G. Konjevod, O. Parekh, A $2\frac{1}{10}$-approximation algorithm for a generalization of the weighted edge-dominating set problem, in: Proc. 8th ESA, 2000, pp. 132–142.

[4] U. Feige, A threshold of $\ln n$ for approximating set cover, in: Proceedings 28th Annual ACM Symposium on the Theory of Computing, 1996, pp. 314–318.

[5] T. Fujito, H. Nagamochi, A 2-approximation algorithm for the minimum weight edge dominating set problem, Discrete Appl. Math., to appear. PII S0166-218X(00)00383-8.

[6] M.R. Garey, D.S. Johnson, The rectilinear Steiner-tree problem is NP-complete, SIAM J. Appl. Math. 32 (4) (1977) 826–834.

[7] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer, Berlin, 1988.

[8] S. Guha, S. Khuller, Approximation algorithms for connected dominating sets, Algorithmica 20 (4) (1998) 374–387.

[9] S. Guha, S. Khuller, Improved methods for approximating node weighted Steiner trees and connected dominating sets, Inform. and Comput., to appear.

[10] M.M. Halldórsson, Approximating the minimum maximal independence number, Inform. Process. Lett. 46 (1993) 169–172.

[11] F. Harary, Graph Theory, Addison-Wesley, Reading, MA, 1969.

[12] R.W. Irving, On approximating the minimum independent dominating set, Inform. Process. Lett. 37 (1991) 197–200.

[13] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.

[14] P. Klein, R. Ravi, A nearly best-possible approximation algorithm for node-weighted Steiner trees, J. Algorithms 19 (1) (1995) 104–115.

[15] J. Koenemann, G. Konjevod, O. Parekh, A. Sinha, Improved approximations for tour and tree covers, in: Proc. APPROX 2000.

[16] S. Rajagopalan, V.V. Vazirani, On the bidirected cut relaxation for the metric Steiner tree problem, in: Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms, 1999, pp. 742–751.

[17] G. Robins, A. Zelikovsky, Improved Steiner tree approximation in graphs, in: Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 770–779.

[18] C. Savage, Depth-first search and the vertex cover problem, Inform. Process. Lett. 14 (5) (1982) 233–235.

[19] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem, Combinatorica 2 (4) (1982) 385–393.

[20] M. Yannakakis, F. Gavril, Edge dominating sets in graphs, SIAM J. Appl. Math. 38 (3) (1980) 364–372.