Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT
Information Processing Letters 89 (2004) 75–83
Information Processing Letters
www.elsevier.com/locate/ipl

Hardness results and approximation algorithms of k-tuple domination in graphs ☆

Ralf Klasing [a,1], Christian Laforest [b,*]

[a] *MASCOTTE project, I3S-CNRS/INRIA/Université de Nice-Sophia Antipolis, 2004 Route des Lucioles, BP 93, F-06902 Sophia Antipolis Cedex, France*

[b] *Tour Evry 2, LaMI, Université d'Evry, 523 place des terrasses, 91000 Evry, France*

Received 26 July 2002; received in revised form 29 September 2003

Communicated by S. Albers

Keywords: Approximation algorithms; Combinatorial problems; Domination problems

1. Introduction

In a graph G, a vertex is said to *dominate* itself and all of its neighbors. A *dominating set* of $G = (V, E)$ is a subset D of V such that every vertex in V is dominated by at least one vertex in D. Domination and its variations have many applications, and have been extensively studied in the literature, see [4,8,9].

Among the variations of domination, the k-tuple domination problem was introduced in [7,8]. For a fixed positive integer k, a *k-tuple dominating set* of $G = (V, E)$ is a subset D_k of V such that every vertex in V is dominated by at least k vertices of D. The special case when $k = 1$ is the usual domination. The case when $k = 2$ is called *double domination* in [7] where exact values of the double domination number of some special graphs are obtained. The same paper also gives various bounds of double and k-tuple domination in terms of other parameters.

A main application to network purposes of k-tuple domination is for *fault tolerance* or *mobility* in the following situations. Each vertex of the graph models a node of the network and edges are links. Node u can use a service (any read-only database for example) only if it is replicated on u or on a neighbor of u. To ensure a certain degree of fault tolerance or to tolerate mobility of nodes, one can imagine that any node u has in its (closed) neighborhood at least k copies of this service available. As each copy can cost a lot, the number of duplicated copies has to be minimized. This is the problem we study.

The purpose of this paper is to study the complexity of the k-tuple domination problem in graphs. The complexity of the (single) domination problem has been well-studied in the literature, see [4]. The hardness of approximation of the (single) domination problem has also been extensively investigated in the literature, see [3]. In terms of the complexity of the k-tuple domination problem in graphs, a linear-time algorithm for the 2-tuple domination problem in trees is given

☆ Work supported by the European projects APPOL, RTN ARACNE and IST FET CRESCCO.

* Corresponding author.

E-mail addresses: Ralf.Klasing@sophia.inria.fr (R. Klasing), laforest@lami.univ-evry.fr (C. Laforest).

[1] This work was done while the first author was visiting the *LaMI* at the Université d'Evry.

doi:10.1016/j.ipl.2003.10.004

in [11]. A linear-time algorithm for the k-tuple domination problem in strongly chordal graphs is presented in [12], where it is also proved that k-tuple domination is NP-complete for split graphs and for bipartite graphs.

In this paper, we extend these studies by investigating the approximation hardness of k-tuple domination in graphs. We also propose several approximation algorithms. Note that an approximation algorithm is a *polynomial time* algorithm that outputs a solution whose cost (here, the size of the dominating set) can be compared to the optimal one. The ratio (more precisely, the worst-case ratio over all input instances) between these two costs is called the *approximation ratio* (not necessarily a constant). General references on approximation algorithms can be found in [3,10].

To our knowledge, this aspect has not been considered before for the k-tuple domination problem. We derive the following results:

(1) We describe a $(\ln|V| + 1)$-approximation algorithm for the k-tuple domination problem in general graphs, and show that k-tuple domination cannot be approximated within a ratio of $(1-\varepsilon)\ln|V|$ for any $\varepsilon > 0$ unless NP $\subseteq$ DTIME$(|V|^{O(\log\log|V|)})$.
(2) We prove that the k-tuple domination problem can be approximated within a constant ratio if the degree of the graph is bounded by a constant, but that it is APX-hard to approximate for graphs of maximum degree $k+2$. Note that a graph G possesses a k-tuple dominating set if and only if the degree of each vertex in G is at least $k-1$.
(3) We show that the k-tuple domination problem can be approximated within a constant ratio in p-claw free graphs, but that it is APX-hard to approximate for 5-claw free graphs. p-claw free graphs are graphs which do not have $K_{1,p}$ (a star with p leaves) as an induced subgraph.

The paper is organized as follows. Section 2 is devoted to the definitions. The following sections present the results on the complexity of the k-tuple domination problem in general graphs (Section 3), bounded degree graphs (Section 4) and p-claw free graphs (Section 5).

2. Definitions

Let $G = (V, E)$ be any undirected graph. We denote by uv an edge of E between u in v. The *neighborhood* of u in G is:

$N_G(u) = \{v\colon\ uv \in E\}$.

The *closed neighborhood* of u in G is:

$N_G[u] = N_G(u) \cup \{u\}$.

The *degree* of any vertex u is $|N_G(u)|$ and the *minimum degree of* G is denoted by δ_G. If $S \subseteq V$ then $G - S$ denotes the graph induced by vertices of $V - S$.

Definition 1. Let $G = (V, E)$ be any undirected graph. Vertex u is *dominated* by vertex v if $u \in N_G[v]$. Vertex u is *k-dominated* if it is dominated by at least k vertices. A *k-tuple domination set* $S \subseteq V$ of G is a set of vertices such that each vertex $u \in V$ is k-dominated by vertices of S. A *minimum* (*or optimal*) *k-tuple domination set of* G is a k-tuple domination set of minimum size.

Problem 1 (*k-tuple domination*). Given a graph G and a constant k, construct a minimum k-tuple domination set of G.

Definition 2. S is an *independent set* of a graph $G = (V, E)$ if $S \subseteq V$ and $uv \notin E$ for all $u, v \in S$. A *maximal independent set* (MIS) S of G is an independent set maximal for inclusion.

It is easy to see that any MIS (Maximal Independent Set) of a graph G is a 1-tuple dominating set (also called *dominating set of G*). Moreover, constructing a MIS can be done by a greedy (linear) algorithm.

Definition 3. A graph G is *p-claw free* if for any vertex u, the graph induced by $N_G(u)$ does not contain an independent set of p vertices.

Equivalently, G is a p-claw free graph if G does not have a $K_{1,p}$ as an induced subgraph.

It is easy to see that G contains a k-tuple dominating set if and only if $\delta_G \geqslant k-1$. In the rest of the paper, k is a constant.

3. k-tuple dominating sets for general graphs

In this section we study Problem 1 in the general case. In Section 3.1 we give a lower bound on the ratio of any approximation algorithm for this problem. In Section 3.2 we propose an approximation algorithm and we prove its approximation ratio.

3.1. Lower bounds on approximation ratio

To formulate our result, we formalize the considered problems as follows.

MIN DOM SET.

Instance: Graph $G = (V, E)$.
Solution: A dominating set of G, i.e., a subset $V' \subseteq V$ such that for all $u \in V - V'$ there is a $v \in V'$ for which $uv \in E$.
Measure: Cardinality of the dominating set, i.e., $|V'|$.

MIN k-TUPLE DOM SET.

Instance: Graph $G = (V, E)$. Constant $k \geqslant 2$.
Solution: A k-tuple dominating set of G, i.e., a subset $V' \subseteq V$ such that each vertex $u \in V$ is k-dominated by vertices of V'.
Measure: Cardinality of the k-tuple dominating set, i.e., $|V'|$.

Theorem 1. *If there is some $\varepsilon > 0$ such that a polynomial time algorithm can approximate* MIN k-TUPLE DOM SET *within* $(1 - \varepsilon)\ln|V|$ *then* NP $\subseteq$ DTIME$(|V|^{O(\log\log|V|)})$.

Proof. We will define an approximation preserving reduction from MIN DOM SET to MIN k-TUPLE DOM SET. This together with the non-approximability bound of MIN DOM SET from [6] (obtained by an explicit transformation of the covering problem studied in [6]) will yield the desired result.

We now describe the reduction from MIN DOM SET to MIN k-TUPLE DOM SET. Given a graph $G = (V, E)$ construct a graph $G_k = (V_k, E_k)$ as follows. We add $k - 1$ vertices $v^1, v^2, \ldots, v^{k-1}$ to G. We connect each vertex $v \in V$ to each of the vertices $v^1, v^2, \ldots, v^{k-1}$, and each vertex v^i is connected to each vertex v^j, for $i \neq j$ ($\{v^1, v^2, \ldots, v^{k-1}\}$ induces a complete graph).

Let D be any dominating set in G. Then $D_k := D \cup \{v^1, v^2, \ldots, v^{k-1}\}$ is a k-tuple dominating set in G_k of size $|D_k| \leqslant |D| + k - 1 \leqslant (1 + k/|D|)|D|$. On the other hand, let D_k be a k-tuple dominating set in G_k. Then $D := D_k - \{v^1, v^2, \ldots, v^{k-1}\}$ is a dominating set in G of size $|D| \leqslant |D_k|$.

Assume that MIN k-TUPLE DOM SET can be approximated within ratio α_k by using an algorithm A_k. Let l be a positive integer. Consider the following algorithm:

Algorithm $A_{k,l}$.
Input: A graph $G = (V, E)$.

1. If a minimum dominating set D of G of size $< l$ exists construct it Else:
2. Compute G_k.
3. Compute a k-tuple dominating set D_k in G_k using Algorithm A_k.
4. Compute $D := D_k - \{v^1, v^2, \ldots, v^{k-1}\}$.
5. Output D.

This algorithm runs in polynomial time since A_k is polynomial and Step 1 is also polynomial (because l is a constant). Note that if D is constructed in line 1 then it is optimal. In the following, we will analyze the case where D is constructed in the next lines.

Let D_k^* be an optimal k-tuple dominating set in G. Let D^* be an optimal (1-tuple) dominating set in G. Note that in our current analysis we have: $|D^*| \geqslant l$. Given graph $G = (V, E)$, Algorithm $A_{k,l}$ computes a dominating set D of G of size

$$
\begin{aligned}
|D| &\leqslant |D_k| \leqslant \alpha_k \left|D_k^*\right| \\
&\leqslant \alpha_k \left(1 + \frac{k}{|D^*|}\right)|D^*| \\
&\leqslant \alpha_k \left(1 + \frac{k}{l}\right)|D^*|.
\end{aligned}
$$

Hence, Algorithm $A_{k,l}$ approximates MIN DOM SET within ratio $\alpha_k(1 + k/l)$. Assume that there is some (fixed) $\varepsilon > 0$ such that MIN k-TUPLE DOM SET can be approximated within ratio $\alpha_k = (1 - \varepsilon)\ln(|V|)$ by using an algorithm A_k. Let l be a positive integer such that: $k/l < \varepsilon/2$. Then Algorithm $A_{k,l}$ approximates MIN DOM SET within ratio

$$\alpha_k\left(1+\frac{k}{l}\right) \leqslant (1-\varepsilon)(1+\varepsilon/2)\ln\bigl(|V|\bigr)$$
$$= (1-\varepsilon')\ln\bigl(|V|\bigr)$$

for $\varepsilon' = \varepsilon/2 + \varepsilon^2/2$. As if MIN DOM SET can be approximated within a ratio of $(1-\varepsilon')\ln|V|$ then NP $\subseteq$ DTIME$(|V|^{O(\log\log|V|)})$ [6], it follows that if MIN k-TUPLE DOM SET can be approximated within a ratio of $(1-\varepsilon)\ln|V|$ then NP $\subseteq$ DTIME$(|V|^{O(\log\log|V|)})$. □

3.2. *Upper bounds on approximation ratio*

To solve Problem 1, we introduce another problem and we make the final correspondence in Theorem 3 at the end of this section.

Definition 4. Let X be any set and $\mathcal{F}$ be any family of subsets of X.

- An element $x \in X$ is *k-covered* in a set $\mathcal{C} \subseteq \mathcal{F}$ of subsets of X if x is in at least k sets of $\mathcal{C}$.
- A *k-cover* of $(X, \mathcal{F})$ is a subset $\mathcal{C}$ of $\mathcal{F}$ such that for all $x \in X$, x is k-covered in $\mathcal{C}$.

Note that when $k = 1$, this problem is the *minimal set cover* problem. A well-known approximation algorithm for this problem can be found in [5]. We generalize it in a sense.

Problem 2 (*Minimal k-cover set*). Let X be any set. Let $\mathcal{F}$ be any family of subsets of X and k be any integer, $k \geqslant 1$. Construct a k-cover of $(X, \mathcal{F})$ of minimum cardinality, called *minimal k-cover set* of $(X, \mathcal{F})$.

Algorithm GEN-SET-COVER.

Input: A set X, a family $\mathcal{F}$ of subsets of X such that $\mathcal{F}$ is a k-cover of $(X, \mathcal{F})$.

1. $\mathcal{C} := \emptyset$;
2. $i := 0$;
3. While $X - S'_1 \cup \cdots \cup S'_i \neq \emptyset$ do
4. $\quad i{+}{+}$;
5. $\quad$ Choose $S \in \mathcal{F} - \mathcal{C}$ such that $|(S - S'_1 \cup \cdots \cup S'_{i-1})|$ is maximized;
6. $\quad S_i := S$;
7. $\quad S'_i :=$ set of elements of X k-covered in $\mathcal{C} \cup \{S_i\}$ but not in $\mathcal{C}$;
8. $\quad \mathcal{C} := \mathcal{C} \cup \{S_i\}$;
9. Output $\mathcal{C}$.

Lemma 1. *Algorithm* GEN-SET-COVER *outputs a k-cover $\mathcal{C}$ of $(X, \mathcal{F})$ in polynomial time.*

Proof. At each step, a *new* element of $\mathcal{F}$ is added to the current solution $\mathcal{C}$. As $\mathcal{F}$ is a k-cover of $(X, \mathcal{F})$, the algorithm will eventually terminate with a k-cover of $(X, \mathcal{F})$. Hence, the number of steps is bounded by $|\mathcal{F}|$ that is polynomial. Each step is also polynomial with appropriate data structures. □

Lemma 2. *Each $x \in X$ is in exactly k sets $(S_i - S'_1 \cup \cdots \cup S'_{i-1})$.*

Proof. By Lemma 1, each $x \in X$ is k-covered in $\mathcal{C}$. So there exists *at least* k subsets S_i such that $x \in (S_i - S'_1 \cup \cdots \cup S'_{i-1})$. Moreover, when $x \in X$ is k-covered for the first time, say at step l, it is included in S'_l and then cannot be in any subsequent $(S_i - S'_1 \cup \cdots \cup S'_{i-1})$ with $i > 1$. □

Notation 1. Based on notations used in the description of the algorithm we define the following sequences:

- For all $i = 1, \ldots, |\mathcal{C}|$,
$$u_i = \frac{1}{|(S_i - S'_1 \cup \cdots \cup S'_{i-1})|}.$$
- For all $x \in X$, let $s_1(x) \leqslant \cdots \leqslant s_k(x)$ be the k indices such that $x \in (S_{s_i(x)} - S'_1 \cup \cdots \cup S'_{s_i(x)-1})$. Let
$$d_x^{s_i(x)} = \frac{1}{|(S_{s_i(x)} - S'_1 \cup \cdots \cup S'_{s_i(x)-1})|},$$
$$c'_x = \sum_{i \in \{s_1(x), \ldots, s_k(x)\}} d_x^i,$$
$$c_x = \max_{i \in \{s_1(x), \ldots, s_k(x)\}} d_x^i.$$

Lemma 3. *Let x be any element of X.*

$$d_x^{s_1(x)} \leqslant \cdots \leqslant d_x^{s_k(x)} = c_x,$$
$$c'_x \leqslant k c_x.$$

Proof. Let us first prove that the sequence $(u_i, i = 1, \ldots, |\mathcal{C}|)$ is increasing. Suppose that for one i, $u_{i+1} < u_i$. This means that

$$\bigl|(S_i - S'_1 \cup \cdots \cup S'_{i-1})\bigr| < \bigl|(S_{i+1} - S'_1 \cup \cdots \cup S'_i)\bigr|.$$

But,

$$\bigl|(S_{i+1} - S'_1 \cup \cdots \cup S'_i)\bigr| < \bigl|(S_{i+1} - S'_1 \cup \cdots \cup S'_{i-1})\bigr|.$$

In this case, at step i the algorithm would have chosen S_{i+1} instead of S_i. This is a contradiction.

Hence, we have proved that for all $x \in X$, the sequence $(d_x^{s_i(x)}, i = 1, \ldots, k)$ increases and by notation, $c_x = d_x^{s_k(x)}$ and $c'_x \leqslant k c_x$. □

Lemma 4.

$$|\mathcal{C}| \leqslant k \sum_{x \in X} c_x.$$

Proof. For any set $Y \neq \emptyset$, we have $\sum_{x \in Y} \frac{1}{|Y|} = 1$. Hence, for any $S_i \in \mathcal{C}$ we have:

$$\sum_{x \in (S_i - S'_1 \cup \cdots \cup S'_{i-1})} \frac{1}{|(S_i - S'_1 \cup \cdots \cup S'_{i-1})|} = 1.$$

Thus,

$$|\mathcal{C}| = \sum_{i=1}^{|\mathcal{C}|} \sum_{x \in (S_i - S'_1 \cup \cdots \cup S'_{i-1})} \frac{1}{|(S_i - S'_1 \cup \cdots \cup S'_{i-1})|}.$$

As any $x \in X$ is in exactly k sets $(S_i - S'_1 \cup \cdots \cup S'_{i-1})$ (Lemma 2), we can rearrange the sum to get:

$$\sum_{x \in X} \sum_{i=1}^{k} \frac{1}{|(S_{s_i(x)} - S'_1 \cup \cdots \cup S'_{s_i(x)-1})|} = \sum_{x \in X} c'_x.$$

Hence we proved that: $|\mathcal{C}| = \sum_{x \in X} c'_x$. The rest of the proof is evident from the obtained result combined with Lemma 3. □

Lemma 5.

$$|\mathcal{C}| \leqslant \sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x.$$

Proof. Let us first prove that:

$$\sum_{x \in X} c_x \leqslant \frac{1}{k} \sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x.$$

For each element $x \in X$, c_x is counted exactly once in $\sum_{x \in X} c_x$. However, as $\mathcal{C}^*$ is a k-cover, c_x is counted at least k times (in the (at least) k sets of $\mathcal{C}^*$ containing x).

The end of the proof is direct with Lemma 4. □

Let p be any positive integer then $H(p)$ is the *harmonic number* of rank p: $H(p) = \sum_{i=1}^{p} \frac{1}{i}$. It is well known (see [5]) that $H(p) \leqslant \ln(p) + 1$.

Lemma 6. *For any $S \in \mathcal{F}$,*

$$\sum_{x \in S} c_x \leqslant H\bigl(|S|\bigr).$$

Proof. Let $z_i = |(S - S'_1 \cup \cdots \cup S'_i)|$ be a sequence, $i = 1, \ldots, |\mathcal{C}|$. $z_0 = |S|$. Let l be the smallest index such that $z_l = 0$ (such l exists because every element of S will be k-covered at the end of the algorithm). Clearly $z_{i-1} \geqslant z_i$. At each step i of the algorithm $z_{i-1} - z_i$ elements of S are k-covered for the first time (note that we can have $z_{i-1} = z_i$). The final cost that an element x gets (in step $s_k(x)$) is $d_x^{s_k(x)}$ (see Lemma 3). Hence,

$$\sum_{x \in S} c_x = \sum_{i=1}^{l} (z_{i-1} - z_i) \frac{1}{|(S_i - S'_1 \cup \cdots \cup S'_{i-1})|}.$$

Because of the maximum choice of the algorithm at each step, we have:

$$\begin{aligned} &\bigl|(S_i - S'_1 \cup \cdots \cup S'_{i-1})\bigr| \\ &\quad \geqslant \bigl|(S - S'_1 \cup \cdots \cup S'_{i-1})\bigr| = z_{i-1}. \end{aligned}$$

It follows:

$$\sum_{x \in S} c_x \leqslant \sum_{i=1}^{l} (z_{i-1} - z_i) \frac{1}{z_{i-1}}.$$

It is proved in [5] that for all integer $a < b$ we have:

$$H(b) - H(a) \geqslant (b - a)\frac{1}{b}.$$

Using this inequality, we obtain:

$$\begin{aligned} \sum_{x \in S} c_x &\leqslant \sum_{i=1}^{l} \bigl(H(z_{i-1}) - H(z_i)\bigr) \\ &= \bigl(H(z_0) - H(z_l)\bigr) = H\bigl(|S|\bigr). \end{aligned}$$ □

Theorem 2. *Let X be any set and $\mathcal{F}$ any family of subsets of X. Let S_M be one set of $\mathcal{F}$ with the maximum cardinality. Then the size of the output of Algorithm* GEN-SET-COVER *with input $(X, \mathcal{F})$ is at most $\ln(|S_M|) + 1$ times the size of the optimal k-cover set of $(X, \mathcal{F})$.*

Proof. From Lemma 5, combined with Lemma 6, we obtain:

$$\begin{aligned}|\mathcal{C}| &\leqslant \sum_{S\in\mathcal{C}^*}\sum_{x\in S} c_x \leqslant \sum_{S\in\mathcal{C}^*} H\big(|S|\big)\\ &\leqslant \sum_{S\in\mathcal{C}^*} H\big(|S_M|\big) = |\mathcal{C}^*| H\big(|S_M|\big)\\ &\leqslant |\mathcal{C}^*|\big(\ln\big(|S_M|\big)+1\big). \qquad \square\end{aligned}$$

Theorem 3. *The minimum k-tuple domination problem in any graph $G = (V, E)$ with maximum degree Δ can be approximated with an approximation ratio of $\ln(\Delta + 1) + 1$.*

Proof. It is easy to see that G contains a minimum k-tuple domination set iff $\delta_G \geqslant k - 1$. If it is the case, let: $(X, \mathcal{F})$ with $X = V$ and $\mathcal{F} = \{N_G[u]\colon u \in V\}$. Apply the algorithm GEN-SET-COVER on $(X, \mathcal{F})$ to obtain $\mathcal{C}$. Finally, output $D = \{u \in V\colon N_G[u] \in \mathcal{C}\}$ that is a k-tuple dominating set of G. Now, with the notations of Theorem 2, S_M corresponds to the maximum size closed neighborhood, that contains at most $\Delta + 1$ vertices. The result follows. $\square$

4. k-tuple domination in bounded-degree graphs

In this section, we show that k-tuple domination can be approximated within a constant ratio if the degree of the graph is bounded by a constant. On the other hand, we show that k-tuple domination is APX-complete (and therefore there is no PTAS) even if the degree of the graph is bounded by $k + 2$. Note that this result is almost tight (in terms of the degree of the graph) as a graph G possesses a k-tuple dominating set if and only if $\delta_G \geqslant k - 1$.

4.1. Membership in APX

Theorem 3 shows that when the graph is degree bounded by a constant, the approximation ratio is constant.

4.2. APX*-completeness*

We first recall the notion of L-reduction (see e.g. [3]).

Definition 5 (*L-reduction*). Given two NP optimization problems F and G and a polynomial transformation f from instances of F to instances of G, we say that f is an *L-reduction* if there are positive constants α and β such that for every instance x of F

(1) $\mathrm{opt}_G(f(x)) \leqslant \alpha \mathrm{opt}_F(x)$,
(2) for every feasible solution y of $f(x)$ with objective value $m_G(f(x), y) = c_2$ we can in polynomial time find a solution y' of x with $m_F(f(x), y') = c_1$ such that $|\mathrm{opt}_F(x) - c_1| \leqslant \beta|\mathrm{opt}_G(f(x)) - c_2|$.

To show the APX-completeness of a problem $\mathcal{P} \in$ APX, it is enough to show that there is an L-reduction from some APX-complete problem to $\mathcal{P}$ (see e.g. [3]). To formulate our result, we formalize the considered problems as follows.

MIN DOM SET-B.

Instance: Graph $G = (V, E)$ of degree bounded by B.
Solution: A dominating set of G, i.e., a subset $V' \subseteq V$ such that for all $u \in V - V'$ there is a $v \in V'$ for which $uv \in E$.
Measure: Cardinality of the dominating set, i.e., $|V'|$.

MIN k-TUPLE DOM SET-B.

Instance: Graph $G = (V, E)$ of degree bounded by B. Constant $k \geqslant 2$.
Solution: A k-tuple dominating set of G, i.e., a subset $V' \subseteq V$ such that each vertex $u \in V$ is k-dominated by vertices of V'.
Measure: Cardinality of the k-tuple dominating set, i.e., $|V'|$.

Now, we are ready to state the main result of this section.

Theorem 4. MIN k-TUPLE DOM SET-$(k + 2)$ *is* APX-*complete for any $k \geqslant 2$.*

Proof. MIN DOM SET-3 is known to be APX-complete [1]. We describe an L-reduction f_k from MIN DOM SET-3 to MIN k-TUPLE DOM SET-$(k+2)$. Given a graph $G=(V,E)$ of bounded degree 3 construct a graph $G_k=(V_k,E_k)$ of bounded degree $k+2$ as follows. For each vertex $v \in V$, we add one complete graph of k vertices $G(v)$ and connect v to $k-1$ of the vertices in $G(v)$. Note that the maximum degree of G_k is $k+2$.

It is easy to see that any k-tuple dominating set in G_k is composed of a dominating set in G plus all the vertices in $G(v)$ for any $v \in V$.

As a result, any k-tuple dominating set $D_k \subseteq V_k$ of $G_k = f_k(G)$ can be transformed into a dominating set $D \subseteq V$ of size $|D| = |D_k| - kn$ where $n=|V|$. Then, for any optimal k-tuple dominating set $D_k^* \subseteq V_k$ and any optimal dominating set $D^* \subseteq V$ the following relation holds: $|D^*| = |D_k^*| - kn$. Hence,

$$|D| - |D^*| = |D_k| - \left|D_k^*\right|.$$

On the other hand, given a dominating set $D \subseteq V$, we can construct a k-tuple dominating set $D_k \subseteq V_k$ of $G_k = f_k(G)$ such that $|D_k| = |D| + kn$. Since G has bounded degree 3, we have $|D| \geqslant n/4$ (see [7]). Therefore $|D_k| = |D| + kn \leqslant (4k+1)|D|$. Thus,

$$\left|D_k^*\right| \leqslant (4k+1)|D^*|$$

and we have shown that f_k is an L-reduction with $\alpha = 4k+1$ and $\beta = 1$. $\square$

5. k-tuple domination in p-claw free graphs

In this section we study the k-tuple domination in p-claw free graphs. We show that the problem is APX complete, even in a restricted class of graphs. We also propose an approximation algorithm having a constant approximation ratio (if p is constant).

5.1. Lower bounds on approximation ratio

Theorem 5. MIN k-TUPLE DOM SET *is* APX-*complete even when restricted to all* 5-*claw free graphs of degree bounded by* $k+2$, *for any constant* $k \geqslant 2$.

Proof. As the set of all p-claw free graphs contains the set of all graphs of degree bounded by $p-1$ as a subclass, we directly obtain the APX completeness from Theorem 4. Indeed, the graphs used in the proof are 5-claw free graphs by construction and have degree bounded by $k+2$. We show in Section 5.2 that MIN k-TUPLE DOM SET is APX when restricted to all 5-claw free graphs, for any $k \geqslant 2$. $\square$

5.2. Upper bounds on approximation ratio

Lemma 7. *Let $G=(V,E)$ be any p-claw free graph and k be any constant such that $\delta_G \geqslant k-1$. Let D_k^* be any optimal k-tuple domination of G and S be any MIS of G. Then*:

$$\frac{k|S|}{(p-1)} \leqslant \left|D_k^*\right|.$$

Proof. For all $u \in S$, let $c_u = |D_k^* \cap N_G[u]|$. As D_k^* is a k-tuple domination of G, $c_u \geqslant k$ for each $u \in S$ and we have:

$$\sum_{u \in S} c_u \geqslant k|S|.$$

For all $v \in D_k^*$, let $d_v = |S \cap N_G[v]|$. As G is a p-claw free graph, for all $v \in D_k^*$ there are at most $p-1$ independent vertices in its neighborhood and $d_v \leqslant p-1$. We have:

$$(p-1)\left|D_k^*\right| \geqslant \sum_{v \in D_k^*} d_v.$$

Now, for all $u \in S$, let $c'_u = |D_k^* \cap N_G(u)|$. Hence, for all $u \in D_k^* \cap S$, $c_u = c'_u + 1$ and for all $u \in S - D_k^*$, $c_u = c'_u$. Thus we have:

$$\sum_{u \in S} c_u = \left|D_k^* \cap S\right| + \sum_{u \in S} c'_u.$$

Similarly, for all $v \in D_k^*$, let $d'_v = |S \cap N_G(v)|$. For all $v \in D_k^* \cap S$, $d_v = d'_v + 1$ and for all $v \in D_k^* - S$, $d_v = d'_v$. Thus we have:

$$\sum_{v \in D_k^*} d_v = \left|D_k^* \cap S\right| + \sum_{v \in D_k^*} d'_v.$$

The last equality finishes the proof:

$$\sum_{u \in S} c'_u = \left|\{uv \in E\colon\ u \in S,\ v \in D_k^*\}\right| = \sum_{v \in D_k^*} d'_v. \qquad \square$$

Algorithm TUPLE-DOMINATING-CLAW.

Input: A p-claw free graph $G = (V, E)$, a constant k such that $k - 1 \leqslant \delta_G$.

1. For $i := 1$ to k
2. Construct a MIS S_i in $G - S_1 \cup \cdots \cup S_{i-1}$;
3. $D := S_1 \cup \cdots \cup S_k$;
4. For $i := 2$ to k
5. Let L_i be the set of vertices of S_1 that are not dominated by any vertex of S_i;
6. $S_i' := \emptyset$;
7. For all it $u \in L_i$, if $N_G[u] - D \neq \emptyset$ choose any vertex $v \in N_G[u] - D$ and add it in D and in S_i';
8. For each vertex $u \in S_2 \cup \cdots \cup S_k$, add in D a sufficient number of new vertices of $N_G[u] - D$ to ensure that u is k-dominated by vertices of D;
9. Output D;

Note that sets S_i' are not used in the algorithm (can be removed) but are useful for the analysis.

Theorem 6. *Algorithm* TUPLE-DOMINATING-CLAW *is a* $\frac{(p-1)}{2}(k - 1 + \frac{2}{k})$ *approximation algorithm for the optimal k-tuple dominating set problem in p-claw free graphs.*

Proof. Let $G = (V, E)$ be any entry graph, k a given constant, D the set of vertices returned by the algorithm and D_k^* an optimal k-tuple dominating set of G. It is clear that the algorithm outputs D in polynomial time. Let us call S_{k+1} the set of new vertices added in the last part of the algorithm (line 7).

Let us see why D is a k-tuple dominating set of G. For all $u \in G - S_1 \cup \cdots \cup S_k$, at each step i, u is not taken in S_i and thus is dominated by one vertex of S_i. At the end, u is dominated by at least k different vertices of $S_1 \cup \cdots \cup S_k$. Now each $u \in S_1 \cup \cdots \cup S_k$ is k-dominated at the end because of the last steps (producing the S_i' sets to k-dominate S_1 and the set S_{k+1} to k-dominate vertices of $S_2 \cup \cdots \cup S_k$).

We can note that $|L_i| \geqslant |S_i'|$, L_i and S_i are disjoint, S_i' and S_i are disjoint, hence $|S_i \cup S_i'| \leqslant |S_i \cup L_i|$. In addition, as $S_i \cup L_i$ is an independent set of G (because L_i and S_i are independent sets of G and if a vertex of $u \in L_i$ is neighbor of $v \in S_i$ then it dominates it and it is a contradiction with the definition of L_i). Then we can apply Lemma 7 on $S_i \cup L_i$ and obtain for all $i = 2, \ldots, k$,

$$|S_i \cup S_i'| \leqslant |S_i \cup L_i| \leqslant \frac{(p-1)|D_k^*|}{k}. \tag{1}$$

Let us now upper bound $|S_{k+1}|$. For each $i = 2, \ldots, k$ and each vertex $u \in S_2 \cup \cdots \cup S_i$, the algorithm adds at most $k - i$ new vertices in S_{k+1}. Indeed, vertex u is already i-dominated by itself and at least one vertex in S_j for each $1 \leqslant j \leqslant i - 1$. Moreover, as each S_i is a MIS, by Lemma 7 we get:

$$|S_i| \leqslant \frac{(p-1)|D_k^*|}{k}.$$

Hence, we have

$$|S_{k+1}| \leqslant \sum_{i=2}^{k} (k-i)|S_i| \leqslant \frac{(p-1)|D_k^*|}{k} \sum_{i=2}^{k} (k-i).$$

Finally:

$$|S_{k+1}| \leqslant (p-1)\frac{(k-1)(k-2)}{2k} \left|D_k^*\right|. \tag{2}$$

Let us prove now the approximation ratio.

$$|D| = |S_1| + \sum_{i=2}^{k} \left|S_i \cup S_i'\right| + |S_{k+1}|.$$

Using (1) and (2), we get:

$$\begin{aligned}|D| &\leqslant (p-1)\left|D_k^*\right| + (p-1)\frac{(k-1)(k-2)}{2k}\left|D_k^*\right| \\ &= \frac{(p-1)}{2}\left(k - 1 + \frac{2}{k}\right)\left|D_k^*\right|. \qquad \square\end{aligned}$$

6. Conclusion and perspectives

We have proved the hardness of the k-tuple domination problem, even in restricted families of graphs. Moreover, we have proposed approximation algorithms for all these families of hard instances. For each of them, the lower and upper bounds on approximation ratios are tight.

The main family studied here are p-claw free graphs in which MIS is a good approximated dominating set (we used this fact intensively in Section 5). We can note that unit disk graphs are 6-claw free graphs (see [13]). Our algorithms can then be applied on this family. However, hardness results for k-tuple domination on unit disk graphs are still unknown (to our knowledge). This would be particularly interesting since unit disk graphs model ad-hoc networks [2].

Acknowledgements

The authors thank Odile Favaron (LRI Orsay) for pointing out reference [7] and Gerard Chang (National Taiwan University) for pointing out references [4,11,12].

References

[1] P. Alimonti, V. Kann, Hardness of approximating problems on cubic graphs, Theoret. Comput. Sci. 237 (2000) 123–134.
[2] K. Alzoubi, P.-J. Wang, O. Frieder, Distributed heuristics for connected dominating sets in wireless ad hoc networks, J. Comm. Networks 4 (2002) 1–8.
[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation, Springer, Berlin, 1999.
[4] G.J. Chang, Algorithmic aspects of domination in graphs, in: D.-Z. Du, P.M. Pardalos (Eds.), in: Handbook of Combinatorial Optimization, Vol. 3, Kluwer Academic Publishers, Dordrecht, 1998, pp. 339–405.
[5] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.
[6] U. Feige, A threshold of $\ln n$ for approximating set cover, J. ACM 45 (1998) 634–652.
[7] F. Harary, T. Haynes, Double domination in graphs, Ars Combin. 55 (2000) 201–213.
[8] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, Domination in Graphs: Advanced Topics, Marcel Dekker Inc., New York, 1998.
[9] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, Domination in Graphs: The Theory, Marcel Dekker Inc., New York, 1998.
[10] D. Hochbaum, Approximation Algorithms for NP-Hard Problems, PWS Publishing Company, Boston, MA, 1997.
[11] C.-S. Liao, G.J. Chang, Algorithmic aspects of k-tuple domination in graphs, Taiwanese J. Math. 6 (2002) 415–420.
[12] C.-S. Liao, G.J. Chang, k-tuple domination in graphs, Inform. Process. Lett. 87 (2003) 45–50.
[13] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, D.J. Rosenkrantz, Simple heuristics for unit disk graphs, Networks 25 (1995) 59–68.