On the approximability and exact algorithms for vector domination and related problems in graphs

Ferdinando Cicalese[a], Martin Milanič[b,c], Ugo Vaccaro[a]

[a] *Dipartimento di Informatica, University of Salerno, 84084 Fisciano (SA), Italy*

[b] *University of Primorska, UP IAM, Muzejski trg 2, SI6000 Koper, Slovenia*

[c] *University of Primorska, UP FAMNIT, Glagoljaška 8, SI6000 Koper, Slovenia*

Abstract

We consider two graph optimization problems called vector domination and total vector domination. In vector domination one seeks a small subset S of vertices of a graph such that any vertex outside S has a prescribed number of neighbors in S. In total vector domination, the requirement is extended to all vertices of the graph. We prove that these problems (and several variants thereof) cannot be approximated to within a factor of $c \ln n$, where c is a suitable constant and n is the number of the vertices, unless $\mathsf{P} = \mathsf{NP}$. We also show that two natural greedy strategies have approximation factors $\ln \Delta + O(1)$, where Δ is the maximum degree of the input graph. We also provide exact polynomial time algorithms for several classes of graphs. Our results extend, improve, and unify several results previously known in the literature.

Keywords: vector domination, total vector domination, α-domination, k-domination, multiple domination, inapproximability, approximation algorithm, polynomial time algorithm, trees, threshold graphs, P_4-free graphs

1. Introduction

The concept of domination in graphs has been extensively studied, both in structural and algorithmic graph theory, because of its numerous applications to a variety of areas. Informally, a set of vertices of a graph is said to dominate a vertex if it contains a sufficient part of its closed neighborhood, where the exact definition of "sufficient" depends on the model. Generally, one seeks small sets that dominate the whole graph. Domination naturally arises in facility location

Email addresses: `cicalese@dia.unisa.it` (Ferdinando Cicalese), `martin.milanic@upr.si` (Martin Milanič), `uv@dia.unisa.it` (Ugo Vaccaro)

problems, in problems involving finding sets of representatives, in monitoring communication or electrical networks, and in land surveying. The two books [21] [22] discuss the main results and applications of domination in graphs. Many variants of the basic concepts of domination have appeared in the literature. Again, we refer to [21] [22] for a survey of the area.

In this paper we provide hardness results, approximation algorithms and exact polynomial time algorithms for an interesting generalization of the basic concept of domination, firstly introduced in [19]. Here, a subset of vertices S is said to dominate a vertex v if either $v \in S$, or there are in S a *prescribed* number of neighbors of v (see below for formal definitions). Again, one seeks small subsets that dominate (in this new sense) the whole vertex set of the graph.

Main Definitions. For a graph $G = (V, E)$ and a vertex $v \in V$, denote by $N(v)$ (or $N_G(v)$, if the graph is not clear from the context) the set of neighbors of v (that is, the *(open) neighborhood* of v), by $N[v] := N(v) \cup \{v\}$ (or $N_G[v]$) the *closed neighborhood* of v, by $d(v) = d_G(v)$ the degree of v, and by $\Delta(G)$ the maximum degree of any vertex in G. A *dominating set* in a graph $G = (V, E)$ is a subset of the graph's vertex set such that every vertex not in the set has a neighbor in the set. A *total dominating set* in G is a subset $S \subseteq V$ such that every vertex of the graph has a neighbor in S, that is, for every $v \in V$ there exists a vertex $u \in S$ such that $uv \in E$.

The *vector domination* is the following problem: Given a graph $G = (V, E)$, and a vector $\mathbf{k} = (k_v : v \in V)$ such that for all $v \in V$, $k_v \in \{0, 1, \ldots, d(v)\}$, find a *vector dominating set* (VDS) of minimum size, that is, a set $S \subseteq V$ minimizing $|S|$ and such that $|S \cap N(v)| \geq k_v$ for all $v \in V \setminus S$. Vector dominating sets were introduced in [19], have also appeared in the literature under the name of *threshold ordinary dominating sets* [18], and were recently studied from the parameterized complexity point of view [34]. The *total vector domination* is the problem of finding a minimum-sized *total vector dominating set*, that is, a set $S \subseteq V$ such that $|S \cap N(v)| \geq k_v$ for all $v \in V$. The minimum sizes of vector and total vector dominating sets will be denoted by $\gamma(G, \mathbf{k})$ and $\gamma^t(G, \mathbf{k})$, respectively. If in the definition of total vector domination we replace open neighborhoods with closed ones, we get the so called *multiple domination* problem [27, 28]: Given a graph $G = (V, E)$ and a vector $\mathbf{k} = (k_v : v \in V)$ such that for all $v \in V$, $k_v \in \{0, 1, \ldots, d(v)+1\}$, find a minimum size set $S \subseteq V$ such that for all vertices $v \in V$, it holds that $|N[v] \cap S| \geq k_v$. We will also consider the following special cases of vector domination, total vector domination, and multiple domination:

- For $0 < \alpha \leq 1$, an *α-dominating set* in G is a subset $S \subseteq V$ such that every vertex not in the set has at least an α-fraction of its neighbors in the set, that is, for every $v \in V \setminus S$, it holds that $|N(v) \cap S| \geq \alpha |N(v)|$.
- For $0 < \alpha \leq 1$, a *total α-dominating set* in G is a subset $S \subseteq V$ such that

every vertex has at least an α-fraction of its neighbors in the set, that is, for every $v \in V$, it holds that $|N(v) \cap S| \geq \alpha|N(v)|$.

- For $0 < \alpha \leq 1$, an *α-rate dominating set* in G is a subset $S \subseteq V$ such that every vertex has at least an α-fraction of the members of its closed neighborhood in the set, that is, for every $v \in V$, it holds that $|N[v] \cap S| \geq \alpha|N[v]|$.

By $\gamma(G)$ ($\gamma_\alpha(G)$, $\gamma^t(G)$, $\gamma^t_\alpha(G), \gamma_{\times\alpha}(G)$) we denote the minimum size of a dominating (α-dominating, total dominating, total α-dominating, α-rate dominating) set in G. For a fixed $0 < \alpha \leq 1$, the problem of finding in a given graph a dominating (α-dominating, total dominating, total α-dominating, α-rate dominating) set of minimum size will be referred to simply as the *domination* (α-domination, *total domination*, *total α-domination*, *α-rate domination*). The notion of α-domination was introduced by Dunbar *et al.* [11] and studied further in [6, 7, 15, 16]. The notion of α-rate domination was introduced in 2009 by Gagarin *et al.* [15, 16]. To the best of our knowledge, the notions of total vector domination and total α-domination are new, that is, they are introduced in [3] and in this paper, respectively.

Notice that for every $\alpha > 0$, every α-dominating set is a dominating set, every total α-dominating set is a total dominating set, every vertex cover is an α-dominating set, and every 1-dominating set is a vertex cover. As shown in [11], for graphs of maximum vertex degree at most $1/\alpha$, α-dominating sets coincide with dominating sets, and it can be shown similarly that for such graphs total α-dominating sets coincide with total dominating sets. Moreover, for graphs of maximum vertex degree less than $1/(1-\alpha)$ (where $\alpha < 1$), α-dominating sets coincide with vertex covers [11]. Clearly, the (total) α-domination corresponds to the special case of the (total) vector domination, in which $k_v = \lceil \alpha \cdot d(v) \rceil$ for all $v \in V$. In fact, we shall mainly use α-domination for our inapproximability results, and provide algorithmic results in terms of the more general problem of vector domination.

In Table 1 we summarize definitions of several domination parameters.[1]

Our Results. We first provide two natural greedy algorithms for vector domination and total vector domination in general graphs, having approximation factors of $\ln(2\Delta(G)) + 1$ and $\ln(\Delta(G)) + 1$, respectively. Subsequently, we prove that the above results are essentially best possible, in the sense that both the α-domination and its total variant are inapproximable within a factor of $c \ln n$ for a suitable constant $c > 0$, unless $\mathsf{P} = \mathsf{NP}$. We also obtain better inapproximability bounds under the stronger hypothesis $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. Notice that our inapproximability results are provided for *any* fixed $0 < \alpha < 1$, and this range of values for α is as large as it can be: the 0-domination, 0-total domina-

[1]The notions of *strict α-domination* and *strict total α-domination* were introduced under different names in [3].

Model	Neighborhood	Total / Partial	Inequal.	Threshold type
α-domination [11]	open	partial	$\geq$	fraction $= \alpha$
α-rate domination [15]	closed	total	$\geq$	fraction $= \alpha$
domination [21]	closed	total	$\geq$	uniform, $k_v = 1\ \forall v$
k-domination [13, 14]	open	partial	$\geq$	uniform, $k_v = k\ \forall v$
k-tuple domination [20]	closed	total	$\geq$	uniform, $k_v = k\ \forall v$
k-tuple total domination [23]	open	total	$\geq$	uniform, $k_v = k\ \forall v$
monopoly [33]	closed	total	$\geq$	fraction $= 1/2$
multiple domination [27, 28]	closed	total	$\geq$	non-uniform
partial monopoly [33]	open	partial	$>$	fraction $= 1/2$
positive influence domination [36]	open	total	$\geq$	fraction $= 1/2$
strict α-domination [3]	open	partial	$>$	fraction $= \alpha$
strict total α-domination [3]	open	total	$>$	fraction $= \alpha$
total α-domination [this paper]	open	total	$\geq$	fraction $= \alpha$
total domination [21]	open	total	$\geq$	uniform, $k_v = 1\ \forall v$
total vector domination [3]	open	total	$\geq$	non-uniform
vector domination [19]	open	partial	$\geq$	non-uniform
vertex cover [21]	open	partial	$\geq$	$k_v = d(v)\ \forall v$

Table 1: Definitions of different domination models. The neighborhoods can be either *open* ($N(v)$) or *closed* ($N[v]$); the domination constraint can be either required for all $v \in V$ (*total*) or only for $v \in V \setminus S$ (*partial*); the type of inequality can be either weak ($\geq$) or strict ($>$); the threshold can be either *uniform* ($k_v = k$ for all $v \in V$), *non-uniform* (every $v \in V$ has its own threshold k_v) or expressed as a *fraction* of the size of the (open or closed) neighborhood ($\alpha \cdot N_v$, where $N_v \in \{N(v), N[v]\}$), according to the neighborhood type as specified in the second column. Notice that some of the models can be defined by more than one choice of the parameters.

tion, and 1-total domination problems are trivial, and the 1-domination problem coincides with the 2-approximable vertex cover problem. We also obtain inapproximability results for other problems listed in Table 1 (with the exception of vertex cover), see Section 3 and the summarizing Table 2 in Section 5.

Subsequently, we individuate special classes of graphs for which vector domination and total vector domination can be optimally solved in polynomial time. More specifically, we provide polynomial time algorithms for computing minimum size vector domination sets and total vector domination sets for complete graphs, trees, P_4-free graphs and threshold graphs.

Related Work. The papers [6, 11, 15] provide several bounds for the value of $\gamma_\alpha(G)$ in terms of other graph parameters, while [7] gives a characterization of α-perfect trees. In addition, Dunbar *et al.* [11] give an NP-completeness result for α-domination. This result is extended considerably by our inapproximability results. Conversely, our approximability result for vector domination answers an open problem posed in [15, 16] where the authors suggest to develop algorithms approximating α-domination to a certain degree of precision.

The algorithmic aspect of total vector domination in strongly chordal graphs (a super-class of trees) was studied in [28], where a polynomial time algorithm for that purpose was given. However, the authors of [28] point out that their approach cannot be modified to handle the case of vector domination, and that a new approach is needed.

Strictly related to our results is also the paper [30]. The authors study the hardness of approximating minimum monopolies in graphs [33]. In the language of Table 1, a monopoly corresponds to a (1/2)-rate dominating set, and a partial monopoly to a strict (1/2)-dominating set. Therefore, our inapproximability results for α-rate domination and strict α-domination can be seen as significant extensions of the results of [30] from the case $\alpha = 1/2$ to arbitrary α. It is also worth mentioning a recent paper on the approximability of the majority monopoly problem [29].

Our findings are also relevant to the new area of influence spread in social networks [25], specifically, to positive influence dominating sets (PIDS) in social networks [36]. In our language, PIDS correspond to total α-dominating sets with $\alpha = 1/2$. In [36] it is proved that PIDS is APX-hard. Our hardness results for total α-domination are more general, and also stronger since we prove inapproximability within a logarithmic factor. In the same area, the paper [38] introduced the problem of identifying a minimum set of nodes that could influence a whole network within a time bound d. There, a set of nodes S influences a new node x in one step ($d = 1$) if the majority of neighbors of x is in S. The paper [38] mostly studies hardness results for the case $d = 1$. It is clear that our scenario includes that of [38] (in the case $d = 1$) and corresponds to a more extensive model of influence among nodes, similar to the one considered in [32] for a related but different problem.

2. Approximability results

In this section, we show that vector domination and total vector domination can be approximated in polynomial time by a factor of $\ln(2\Delta(G)) + 1$ and $\ln(\Delta(G)) + 1$, respectively. (We denote by ln the natural logarithm.)

We start with total vector domination and related problems. Our results are based on the results for the set cover problem. Consider the following generalization of the set cover problem:

`SET MULTICOVER`

Instance: A set-system $\mathcal{C} = (U, \mathcal{F})$, where U is a finite ground set and $\mathcal{F}$ is a collection of subsets of U; a non-negative integer requirement $req(u)$ for every element u of the ground set.

Task: Find a minimum size subcollection $\mathcal{F}' \subseteq \mathcal{F}$ such that every element u appears in at least $req(u)$ sets in $\mathcal{F}'$.

The decision version of the `SET MULTICOVER` problem is NP-complete [17]. Moreover, the greedy algorithm produces a solution that is always within a factor $\ln \Delta + 1$ of the optimum, where Δ is the maximum size of a set in $\mathcal{F}$ [8].

Every instance of any of the "total" domination problems defined in Table 1 (see the third column of the table) can be described as an instance of the `SET MULTICOVER` problem. For example, if $(G, \mathbf{k})$ is an instance to the total vector domination problem, we can take $U = V(G)$, define $\mathcal{F}$ to be the collection of all (open) neighborhoods, and set $req(u) = k_u$ for all $u \in U$. It is clear that a subset $S \subseteq V(G)$ is a total vector dominating set for $(G, \mathbf{k})$ if and only if the collection $(N(v) : v \in S)$ is a feasible solution to the instance $(U, \mathcal{F}, req)$ of the `SET MULTICOVER` problem. Similar transformations work for the other "total" domination problems.

We therefore obtain the following results and their corollaries:

Theorem 1. *Total vector domination can be approximated in polynomial time by a factor of* $\ln(\Delta(G)) + 1$.

Corollary 1. *Total α-domination, strict total α-domination, k-tuple total domination, and positive influence domination problems can be approximated in polynomial time by a factor of* $\ln(\Delta(G)) + 1$.

Theorem 2. *The multiple domination problem can be approximated in polynomial time by a factor of* $\ln(\Delta(G) + 1) + 1$.

Corollary 2. *The α-rate domination, the k-tuple domination problem and the monopoly problem can be approximated in polynomial time by a factor of* $\ln(\Delta(G) + 1) + 1$.

The $(\ln(\Delta(G)) + 1)$-approximability of the positive influence domination problem and the $(\ln(\Delta(G) + 1) + 1)$-approximability of the k-tuple domination problem were proved in [36] and [26], respectively.

The above approach does not seem to be easily applicable to "partial" domination problems such as vector domination, α-domination, k-domination, partial monopoly and strict α-domination. Instead, we will show below that these problems can be recast as a particular case of the well known `MINIMUM SUBMODULAR COVER` problem, and apply a classical result due to Wolsey [37].

Theorem 3. *Vector domination can be approximated in polynomial time by a factor of* $\ln(2\Delta(G)) + 1$.

Proof. For a graph $G = (V, E)$ and a vector $\mathbf{k} = (k_v : v \in V)$ s.t. for all

$v \in V$, $k_v \in \{0, 1, \ldots, d(v)\}$, we define a function $f : 2^V \longrightarrow \mathbb{N}$, as follows:

$$\text{for all } S \subseteq V\,, \quad \text{let} \quad f(S) = \sum_{v \in V} \tau_v(S)\,, \quad \text{where}$$
$$\tau_v(S) = \begin{cases} \min\{|S \cap N(v)|, k_v\}, & \text{if } v \notin S; \\ k_v, & \text{if } v \in S. \end{cases} \tag{1}$$

The following properties of f can be verified: (i) f is integer valued; (ii) $f(\emptyset) = 0$; (iii) *f is non-decreasing*; (iv) *A set $S \subseteq V$ satisfies $f(S) = f(V)$ if and only if S is a vector dominating set*; (v) *f is submodular.*

Recall that a function $f : 2^V \longrightarrow \mathbb{N}$ is *submodular* if for all $S \subseteq T \subseteq V$ and for all $w \in V \setminus T$, the inequality $f(T \cup \{w\}) - f(T) \leq f(S \cup \{w\}) - f(S)$ holds. The only non-trivial property to show is (v), i.e, the submodularity of f. The proof is given below.

Lemma 1. *The function $f : 2^V \longrightarrow \mathbb{N}$, given by (1), is submodular.*

Proof. It suffices to show that all the functions $\tau_v(\cdot)$ are submodular, that is, that for all $S \subseteq T \subseteq V$ and for all $w \in V \setminus T$,

$$\tau_v(T \cup \{w\}) - \tau_v(T) \leq \tau_v(S \cup \{w\}) - \tau_v(S)\,. \tag{2}$$

Observe that τ_v is non-decreasing.

Suppose first that $\tau_v(T) = k_v$. Then $\tau_v(T \cup \{w\}) = k_v$ and the left-hand side of inequality (2) is equal to 0. Hence inequality (2) holds since τ_v is non-decreasing.

From now on, we assume that $\tau_v(T) < k_v$, which implies $\tau_v(T) = |T \cap N_G(v)|$. Since τ_v is non-decreasing, $\tau_v(S) < k_v$, and hence $\tau_v(S) = |S \cap N_G(v)|$. Inequality (2) simplifies to

$$\tau_v(T) - \tau_v(S) = |(T \setminus S) \cap N_G(v)| \geq \tau_v(T \cup \{w\}) - \tau_v(S \cup \{w\})\,. \tag{3}$$

We may assume that $\tau_v(T \cup \{w\}) > \tau_v(S \cup \{w\})$, since otherwise the right-hand side of (3) equals 0, and inequality (3) holds.

Therefore, $\tau_v(S \cup \{w\}) < k_v$, implying $\tau_v(S \cup \{w\}) = |(S \cup \{w\}) \cap N_G(v)|$. If also $\tau_v(T \cup \{w\}) < k_v$ then $\tau_v(T \cup \{w\}) = |(T \cup \{w\}) \cap N_G(v)|$ and equality holds in (3).

So we may assume that $\tau_v(T \cup \{w\}) = k_v$. Note that v does not belong to $T \cup \{w\}$ (since otherwise either $\tau_v(T)$ or $\tau_v(S \cup \{w\})$ would equal to k_v). Suppose that the inequality (3) fails. Then

$$|(T \setminus S) \cap N_G(v)| < k_v - |(S \cup \{w\}) \cap N_G(v)|\,,$$

which implies

$$|(T \cup \{w\}) \cap N_G(v)| < k_v \,.$$

However, together with the fact that $v \notin T \cup \{w\}$, this contradicts the assumption that $\tau_v(T \cup \{w\}) = k_v$. □

Back to the proof of Theorem 3, by (iv) we have that an optimal solution to the vector dominating set is provided by a minimum size S such that $f(S) = f(V)$. In other words, we have recast vector domination as a particular case of the `MINIMUM SUBMODULAR COVER` [37].

Let $\mathbb{A}$ denote the natural greedy strategy which starts with $S = \emptyset$ and iteratively adds to S the element $v \in V \setminus S$ s.t. $f(S \cup \{v\}) - f(S)$ is maximum, until $f(S) = f(V)$ is achieved. By a classical result of Wolsey [37], it follows that algorithm $\mathbb{A}$ is a $(\ln(\max_{y \in V} f(\{y\})) + 1)$-approximation algorithm for vector domination. For every $y \in V$, we have $f(\{y\}) = \sum_{v \in V \setminus \{y\}} \tau_v(\{y\}) + \tau_y(\{y\}) \leq d(y) + k_y \leq 2d(y)$. Hence $\max_{y \in V} f(\{y\}) \leq 2\Delta(G)$ yielding the desired result. □

Since α-domination, k-domination, partial monopoly and strict α-domination problems are all special cases of the vector domination problems, Theorem 3 implies the following result:

Corollary 3. *α-domination, k-domination, partial monopoly and strict α-domination problems can be approximated in polynomial time by a factor of* $\ln(2\Delta(G)) + 1$.

3. Inapproximability results

Recall the following result on the inapproximability of domination and total domination, which was derived from the analogous result about the set cover problem due to Feige [12]. Hereafter, n denotes the number of vertices of the input graph.

Theorem 4. *[2] For every $\epsilon > 0$, there is no polynomial time algorithm approximating domination (total domination) for graphs without isolated vertices within a factor of $(1 - \epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

Most of our inapproximability results are given in terms of the variants of the α-domination problem. In fact, it turns out that α-domination, its total variant, and the α-rate domination are inapproximable within a $c \ln n$ factor (for suitable constants c) as shown in Theorems 5, 7 and 9 below. *A fortiori* the same results hold for the vector domination, total vector domination and multiple domination problems. Hence, the approximations results of the previous section are essentially best possible. We shall use the following lemma which is an *ad hoc* extension of the hardness of approximating domination within $(1-\epsilon)\ln n$ given in [2].

Lemma 2. *For every integer $B > 0$ and for every $\epsilon > 0$, there is no polynomial time algorithm approximating domination on input graphs G without isolated vertices satisfying $\gamma(G) \geq B\Delta(G)$ within a factor of $(\frac{1}{2}-\epsilon)\ln n$, unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.

Proof. Let B be a positive integer and $\epsilon \in (0, \frac{1}{2})$. We make a reduction from domination on graphs without isolated vertices (Theorem 4). Let G be a graph without isolated vertices with $|V(G)| \geq B^{1/\epsilon}$ that is an instance to domination. We transform G into a graph G' which consists of $N = B\Delta(G)$ disjoint copies of G, say $G_1, \dots, G_N$. Then clearly $\gamma(G') = N\gamma(G)$, while $\Delta(G') = \Delta(G)$. In particular, since $\gamma(G) \geq 1$, the graph G' satisfies $\gamma(G') \geq N = B\Delta(G')$.

For brevity, let us write $n = |V(G)|$ and $n' = |V(G')|$. Suppose that there exists a polynomial time algorithm A' that computes a $(\frac{1}{2}-\epsilon)\ln n'$-approximation to domination in G'. Let S' be the set computed by A'. Then $|S'| \leq (\frac{1}{2}-\epsilon)(\ln n')\gamma(G')$.

For $i = 1, \dots, N$, let $S'_i = S' \cap V(G_i)$, and let $S = S'_{i^*}$ such that $|S'_{i^*}| \leq |S'_i|$ for all $1 \leq i \leq N$. Then S is a dominating set in (the i^*-th copy of) G. Moreover, we can bound the size of S from above as follows:

$$\begin{aligned}
|S| &\leq (1/N)\cdot|S'| && \text{(by the definition of } S\text{)}\\
&\leq (1/N)\cdot(\tfrac{1}{2}-\epsilon)(\ln n')\cdot\gamma(G') && \text{(by the assumption on } A'\text{)}\\
&= (\tfrac{1}{2}-\epsilon)\ln(B\Delta(G)n)\cdot\gamma(G) && \text{(by the properties of } G'\text{)}\\
&\leq (\tfrac{1}{2}-\epsilon)\ln(n^{2+\epsilon})\cdot\gamma(G) && \text{(since } B \leq n^\epsilon \text{ and } \Delta(G) \leq n\text{)}\\
&= (\tfrac{1}{2}-\epsilon)(2+\epsilon)(\ln n)\cdot\gamma(G)\\
&= (1-\epsilon')(\ln n)\cdot\gamma(G)\,,
\end{aligned}$$

where $\epsilon' := 3/2\epsilon + \epsilon^2 \in (0,1)$. Therefore, if there exists a polynomial time algorithm that computes a $(\frac{1}{2}-\epsilon)\ln n'$-approximation to domination in G', there exists a polynomial time algorithm that computes a $(1-\epsilon')\ln n$-approximation to domination in G, and hence, by Theorem 4, this is only possible if $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. □

Theorem 5. *For every $\alpha \in (0,1)$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating α-domination within a factor of $(\frac{1}{2}-\epsilon)\ln n$, unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.

Proof. Let $0 < \alpha < 1$ and $\epsilon \in (0, \frac{1}{2})$. We define $N = \lceil \frac{\alpha}{1-\alpha} \rceil$ and $B = \lceil N/\epsilon \rceil$. Let G be a graph without isolated vertices, with $\gamma(G) \geq B\Delta(G)$ and such that $|V(G)| \geq (N+1)^{1/\epsilon}$. We transform G into a graph G' which consists of G together with a set K of $k = N\Delta(G)$ vertices such that K is disjoint from

$V(G)$. In addition, every vertex v from G is adjacent to precisely k_v vertices in K, where

$$k_v = \begin{cases} \left\lceil \frac{\alpha d_G(v)-1}{1-\alpha} \right\rceil, & \text{if } d_G(v) \geq 2; \\ 0, & \text{if } d_G(v) = 1. \end{cases}$$

(This assignment is done in an arbitrary way.) Notice that the k_v's are well defined, since G has no isolated vertices. Moreover, if $d_G(v) = 1$ then $k_v = 0 < k$, while for $d_G(v) \geq 2$ we have

$$0 \leq k_v = \left\lceil \frac{\alpha d_G(v) - 1}{1-\alpha} \right\rceil < \frac{\alpha d_G(v) - 1}{1-\alpha} + 1 < \frac{\alpha}{1-\alpha} d_G(v) \leq N\Delta(G) = k\,.$$

Hence it is indeed possible to assign to v precisely k_v neighbors in K.

In addition, k_v is an integer satisfying

$$\frac{k_v}{d_G(v) + k_v} < \alpha \leq \frac{k_v + 1}{d_G(v) + k_v}\,.$$

These inequalities are instrumental to the following result.

Claim: $\gamma(G) \leq \gamma_\alpha(G') \leq \gamma(G) + k\,.$

Proof. Let S' be an optimal α-dominating set in G'. Then, the set $S := S' \cap V(G)$ is a dominating set in G. Indeed, suppose for a contradiction that there exists a vertex v in G such that S misses the closed neighborhood of v. Then $|N_{G'}(v) \cap S'| \leq k_v$. The degree of v in G' is equal to $d_{G'}(v) = d_G(v) + k_v$. Therefore

$$\frac{|N_{G'}(v) \cap S'|}{d_{G'}(v)} \leq \frac{k_v}{d_G(v) + k_v} < \alpha\,,$$

contrary to the assumption that S' is α-dominating. Consequently $\gamma(G) \leq |S' \cap V(G)| \leq |S'| = \gamma_\alpha(G')$.

Conversely, let S be an optimal dominating set in G. The set $S' := S \cup K$ is then an α-dominating set in G' such that $|S'| = \gamma(G) + k$. To see that S' is α-dominating in G', observe that for every $v \in V(G') \setminus S' = V(G) \backslash S$, the set $N_{G'}(v) \cap S'$ is the disjoint union of sets $N_G(v) \cap S$ and $N_{G'}(v) \cap K$. Hence

$$|N_{G'}(v) \cap S'| = |N_G(v) \cap S| + |N_{G'}(v) \cap K| \geq 1 + k_v \geq \alpha(d_G(v) + k_v) = \alpha |N_{G'}(v)|\,,$$

where the second inequality holds by the choice of k_v. Altogether, this shows that $\gamma_\alpha(G') \leq |S'| = \gamma(G) + k$ and completes the proof of the claim.

Again, let us write $n = |V(G)|$ and $n' = |V(G')|$. Notice that we have $N + 1 \leq n^\epsilon$ and hence

$$n' = n + N\Delta(G) \leq n + Nn \leq n^{1+\epsilon}\,.$$

Moreover, $\frac{1}{\epsilon} k = \frac{1}{\epsilon} N\Delta(G) \leq B\Delta(G) \leq \gamma(G)$, which implies $k \leq \epsilon\gamma(G)$.

Suppose that there exists a polynomial time algorithm A' which computes an α-dominating set S' for G' such that $|S'| \leq (\frac{1}{2}-\epsilon)(\ln n')\gamma_\alpha(G')$. Let $S = S' \cap V(G)$. Like in the proof of the above claim, we see that S is a dominating set in G. We can bound the size of S from above as follows:

$$\begin{array}{rcll} |S| & \leq & |S'| & \\ & \leq & (\frac{1}{2}-\epsilon)(\ln n')\cdot\gamma_\alpha(G') & \text{(by the assumption on } A'\text{)} \\ & \leq & (\frac{1}{2}-\epsilon)(\ln n^{1+\epsilon})\cdot(\gamma(G)+k) & \text{(by the Claim and } n' \leq n^{1+\epsilon}\text{)} \\ & \leq & (\frac{1}{2}-\epsilon)(1+\epsilon)(\ln n)\cdot(\gamma(G)+\epsilon\gamma(G)) & \text{(since } k \leq \epsilon\gamma(G)\text{)} \\ & = & (\frac{1}{2}-\epsilon)(1+\epsilon)^2(\ln n)\cdot\gamma(G) & \\ & = & (\frac{1}{2}-\epsilon')(\ln n)\cdot\gamma(G)\,, & \end{array}$$

where $\epsilon' := \epsilon^2(\epsilon+3/2) \in (0,\frac{1}{2})$. Therefore, if there exists a polynomial time algorithm that computes a $(\frac{1}{2}-\epsilon)\ln n'$-approximation to α-domination in G', there exists a polynomial time algorithm that computes a $(\frac{1}{2}-\epsilon')\ln n$-approximation to domination in G, and hence, by Lemma 2, this is only possible if $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. □

With minor modifications of the above proof (adapting it for strict inequalities, cf. Table 1), one can obtain the analogue of Theorem 5 for strict α-domination.

Theorem 6. *For every $\alpha \in (0,1)$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating strict α-domination within a factor of $(\frac{1}{2}-\epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

Theorem 6 is a significant extension of the same result for the case $\alpha = 1/2$ proved in [30]. See also [3] for a proof of the inapproximability of strict α-domination within a factor of the form $c\ln n$ for some constant $c > 0$.

By means of a slightly more involved construction, we now prove a similar result for total α-domination.

Theorem 7. *For every $\alpha \in (0,1)$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating total α-domination within a factor of $(\frac{1}{3}-\epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

Proof. Let $0 < \alpha < 1$ and $\epsilon \in (0,\frac{1}{3})$. Let $B = \lceil\frac{\alpha}{1-\alpha}\rceil$. We make a reduction from total domination on graphs G with n vertices such that

$$n \geq \max\left\{\sqrt{\frac{1-\alpha}{\alpha}}, 2^{3/\epsilon}, (B+1)^{2/\epsilon}\right\}, \tag{4}$$

$$\lceil n^{1+\epsilon/3}\rceil \leq n^{1+2\epsilon/3}, \tag{5}$$

$$n + B \leq n^{1+\epsilon/3}\,. \tag{6}$$

and

$$\gamma^t(G) \geq \frac{B}{\epsilon}\,. \tag{7}$$

Clearly, these assumptions are without loss of generality since the inequalities in (4)–(6) are satisfied for all large enough n, while if the inequality (7) is violated, we can find an optimal solution in polynomial time by verifying all subsets of $V(G)$ of size less than $\frac{B}{\epsilon}$.

Let G be a graph satisfying (4)–(7). Let $n = |V(G)|$ and $m := \lceil n^{1+\epsilon/3} \rceil$. We transform G into a graph G' as follows: G' consists of mn disjoint copies of G, say $G_1, \ldots, G_{mn}$, together with a complete graph K on Bmn vertices such that K is disjoint from the mn copies of G. (See Fig. 1.) To describe the remaining edges, we first partition the vertex set of K into m equally-sized parts $K_1, \ldots, K_m$. (In particular, $|K_i| = Bn$ for all $i = 1, \ldots, m$.) Finally, for every $j \in \{1, \ldots, mn\}$, we make every vertex $v \in V(G_j)$ adjacent to precisely k_v vertices in $K_{\lceil j/n \rceil}$ where k_v is an integer satisfying

$$\frac{k_v}{d_G(v) + k_v} < \alpha \leq \frac{1 + k_v}{d_G(v) + k_v}\,.$$

Similarly as in the proof of Theorem 5, we can take $k_v = \lceil \frac{\alpha d_G(v) - 1}{1-\alpha} \rceil$ if $d_G(v) \geq 2$ and $k_v = 0$ if $d_G(v) = 1$. (The graph G, as input to total domination, does not have any isolated vertices, since otherwise the problem is infeasible.) Also notice that since $k_v \leq Bn$, it is indeed possible to assign to every $v \in V(G_j)$ precisely k_v neighbors in $K_{\lceil j/n \rceil}$. (This assignment is done in an arbitrary way.)

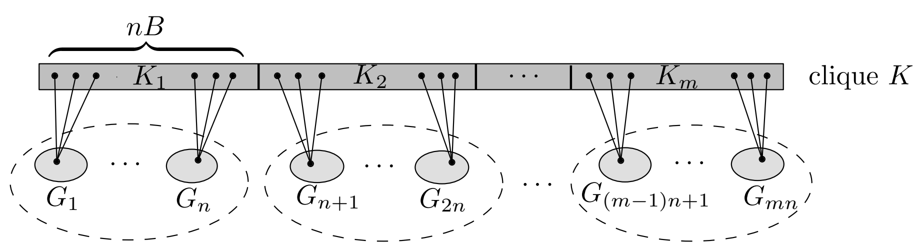

Figure 1: The graph G' in the proof of Theorem 7

Claim: $mn\gamma^t(G) \leq \gamma^t_\alpha(G') \leq mn\gamma^t(G) + Bmn\,.$

Proof of Claim:

Let S' be an optimal total α-dominating set in G', that is, $|S'| = \gamma^t_\alpha(G')$. For every $j = 1, \ldots, mn$, let $S'_j = S' \cap V(G_j)$ denote the part of S' that belongs to to the j-th copy of G in G'. Pick an index $j^* \in \{1, \ldots, mn\}$ for which the size of S'_j is smallest. We argue that the set $S := S_{j^*}$ is a total dominating set in G_{j^*} (and thus in G). Indeed, suppose for contradiction that there exists a vertex v

in G_{j^*} such that S misses the neighborhood of v. Then $|N_{G'}(v) \cap S'| \leq k_v$ while the degree of v in G' is equal to $d_{G'}(v) = d_G(v) + k_v$. Therefore

$$\frac{|N_{G'}(v) \cap S'|}{d_{G'}(v)} \leq \frac{k_v}{d_G(v) + k_v} < \alpha\,,$$

contrary to the assumption that S' is total α-dominating. This implies that $\gamma^t(G) \leq |S|$ and consequently $mn\gamma^t(G) \leq mn|S| \leq \sum_{j=1}^{mn} |S'_j| \leq |S'| = \gamma^t_\alpha(G')$.

Conversely, let S be an optimal total dominating set in G. For $j = 1, \ldots, mn$, let S_j denote the copy of S in G_j, and let $S' = K \cup \bigcup_{j=1}^{mn} S_j$. The set $S' \subseteq V(G')$ satisfies $|S'| = mn\gamma^t(G) + Bnm$. Moreover, S' is a total α-dominating set in G':

- For every $j = 1, \ldots, mn$ and for every $v \in V(G_j)$, the set $N_{G'}(v) \cap S'$ is the disjoint union of sets $N_{G_j}(v) \cap S_j$ and $N_{G'}(v) \cap K$. Hence

  $$|N_{G'}(v) \cap S'| = |N_{G_j}(v) \cap S_j| + |N_{G'}(v) \cap K| \geq$$

  $$\geq 1 + k_v \geq \alpha(d_{G_j}(v) + k_v) = \alpha|N_{G'}(v)|\,.$$

  The second inequality holds by the choice of k_v.

- Let $v \in K$. By construction of G', v is adjacent to every other vertex in K, and to at most n^2 remaining vertices. Hence $d_{G'}(v) \leq (|K|-1)+n^2 = Bmn+n^2-1$. Moreover, $|N_{G'}(v) \cap S'| \geq |K|-1 = Bmn-1$. Therefore, to show that $|N_{G'}(v) \cap S'| \geq \alpha|N_{G'}(v)|$, it suffices to prove that

  $$\frac{Bmn - 1}{Bmn + n^2 - 1} \geq \alpha\,.$$

  This is equivalent to $Bmn(1-\alpha) \geq \alpha n^2 - \alpha + 1$, which (since $n^2 \geq \frac{1-\alpha}{\alpha}$) follows from the inequality $Bmn(1-\alpha) \geq 2\alpha n^2$, or, equivalently,

  $$m \geq \frac{2\alpha}{(1-\alpha)B} n\,,$$

  which follows from the inequality $n^{\epsilon/3} \geq 2$, which, finally, holds true by (4).

This shows that $\gamma^t_\alpha(G') \leq mn\gamma^t(G) + Bmn$ and completes the proof of the claim.

Let us write $n' = |V(G')|$. By assumptions (5) and (6) we have

$$n' = mn^2 + Bmn \leq n^{3+2\epsilon/3} + Bn^{2+2\epsilon/3} = (n+B)n^{2+2\epsilon/3} \leq n^{1+\epsilon/3} n^{2+2\epsilon/3} = n^{3+\epsilon}\,.$$

Suppose that there exists a polynomial time algorithm A' that computes a $(\frac{1}{3} - \epsilon)\ln n'$-approximation to total α-domination in G'. Let S' be the set computed by A'. Then $|S'| \leq (\frac{1}{3} - \epsilon)(\ln n')\gamma^t_\alpha(G')$.

Similarly as in the proof of the claim above, let $S'_j = S' \cap V(G_j)$ and pick an index $j^* \in \{1, \dots, mn\}$ for which the value of $|S'_j|$ is smallest. Then, setting $S = S'_{j^*}$ results in a total dominating set in G_j (and hence in G).

We can bound the size of S from above as follows:

$$\begin{array}{rcll}
|S| & \leq & \frac{1}{mn} \cdot |S'| & \text{(by the choice of } j^*\text{)} \\
& \leq & \frac{1}{mn}(\frac{1}{3} - \epsilon)(\ln n') \cdot \gamma^t_\alpha(G') & \text{(by the assumption on } A'\text{)} \\
& \leq & \frac{1}{mn}(\frac{1}{3} - \epsilon)(\ln(n^{3+\epsilon})) \cdot (mn\gamma^t(G) + Bnm) & \text{(by the Claim and } n' \leq n^{3+\epsilon}\text{)} \\
& = & (\frac{1}{3} - \epsilon)(3 + \epsilon)(\ln n) \cdot (\gamma^t(G) + B) & \\
& \leq & (\frac{1}{3} - \epsilon)(3 + \epsilon)(1 + \epsilon)(\ln n) \cdot \gamma^t(G) & \text{(by (7))} \\
& = & (1 - \epsilon')(\ln n)\gamma^t(G)\,, &
\end{array}$$

where $\epsilon' = \epsilon^3 + 11\epsilon^2/3 + 5\epsilon/3 \in (0, 1)$. Therefore, S approximates the total domination within a factor of $(1 - \epsilon') \ln n$. By Theorem 4, this shows that there is no polynomial time algorithm approximating total α-domination within a factor of $(\frac{1}{3} - \epsilon) \ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log \log n)})$. □

Again, with obvious modifications of the above proof, one can obtain the analogue of Theorem 7 for strict total α-domination.

Theorem 8. *For every $\alpha \in (0, 1)$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating strict total α-domination within a factor of $(\frac{1}{3} - \epsilon) \ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log \log n)})$.*

A different minor modification of the above proof (perform a reduction from domination instead of total domination, and define $k_v = \lceil \frac{\alpha d_G(v) + \alpha - 1}{1 - \alpha} \rceil$ for all $v \in V(G)$) can be used to obtain the analogue of Theorem 7 for α-rate domination:

Theorem 9. *For every $\alpha \in (0, 1)$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating α-rate domination within a factor of $(\frac{1}{3} - \epsilon) \ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log \log n)})$.*

A proof of Theorem 9 can be found in the appendix. Theorem 9 is a significant extension of the result for the case $\alpha = 1/2$ proved in [30]. At the same time, this result complements the $\ln(\Delta(G) + 1)$-approximation algorithm for α-rate domination outlined in Section 2, providing an almost complete answer to the question about the approximability of α-rate domination posed by Gagarin *et al.* in [15, 16].

We continue with inapproximability results for k-domination and k-tuple total domination. With a similar approach to the one by Klasing and Laforest showing the inapproximability of k-tuple domination [26], we obtain the following result.

Theorem 10. *For every $k \geq 1$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating k-domination within a factor of $(1-\epsilon)\ln n$, unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.

Proof. We make a reduction from domination on graphs G with n vertices such that

$$n + k - 1 \leq n^{1+\epsilon} \tag{8}$$

and

$$\gamma(G) \geq \frac{(k-1)(1+\epsilon)}{\epsilon^2}\,. \tag{9}$$

Clearly, this assumption is without loss of generality since the inequality in (8) is satisfied for all large enough n, while if the inequality (9) is violated, we can find an optimal solution in polynomial time by verifying all subsets of $V(G)$ of size less than $\frac{(k-1)(1+\epsilon)}{\epsilon^2}$. Notice that inequality (9) is equivalent to the following inequality:

$$\gamma(G) + k - 1 \leq \frac{1+\epsilon+\epsilon^2}{1+\epsilon} \cdot \gamma(G)\,. \tag{10}$$

Let G be a graph satisfying (8)–(9). We transform G into a graph G' by adding to it a set K of $k-1$ vertices inducing a complete graph such that $V(G)$ and K are disjoint, and connecting every vertex in G to every vertex in K. Let $\gamma^{(k)}(G')$ denote the minimum cardinality of a k-dominating set in G'. For every dominating set S in G, the set $S \cup K$ is a k-dominating set in G'. Hence

$$\gamma^{(k)}(G') \leq \gamma(G) + k - 1\,. \tag{11}$$

Let $n' = |V(G')|$, and suppose that there exists a polynomial time algorithm A' that computes a $(1-\epsilon)\ln n'$-approximation to k-domination in G'. Let S' be the set computed by A'. Then $|S'| \leq (1-\epsilon)(\ln n')\gamma^{(k)}(G')$. Moreover, the set $S := S' \cap V(G)$ is a dominating set in G.

We can bound the size of S from above as follows:

$$\begin{aligned}
|S| &\leq |S'| \\
&\leq (1-\epsilon)(\ln n')\gamma^{(k)}(G') && \text{(by the assumption on } A'\text{)} \\
&\leq (1-\epsilon)(\ln(n+k-1))(\gamma(G)+k-1) && \text{(by (11) and } n' = n+k-1\text{)} \\
&\leq (1-\epsilon)\left(\ln(n^{1+\epsilon})\right)\left(\tfrac{1+\epsilon+\epsilon^2}{1+\epsilon}\gamma(G)\right) && \text{(by (8) and (10))} \\
&= (1-\epsilon')(\ln n)\gamma(G)\,,
\end{aligned}$$

where $\epsilon' = \epsilon^3 > 0$. Therefore, S approximates domination within a factor of $(1-\epsilon')\ln n$. By Theorem 4, this shows that there is no polynomial time algorithm approximating k-domination within a factor of $(1-\epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. □

A minor modification of the above proof (perform a reduction from total domination instead of domination) can be used to obtain the analogue of Theorem 10 for k-tuple total domination.

Theorem 11. *For every $k \geq 1$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating k-tuple total domination within a factor of $(1-\epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

3.1. Inapproximability under the $\mathsf{P} \neq \mathsf{NP}$ assumption

Alon *et al.* proved in [1] that there is no polynomial time algorithm approximating the set cover problem within a factor of $c \ln n$ for some constant $c > 0.2267$, unless $\mathsf{P} = \mathsf{NP}$. Thus, one could obtain an analogue of Theorem 4 with a weaker inapproximability bound and under the assumption that $\mathsf{P} \neq \mathsf{NP}$.

In particular, with a straightforward adaptation of the proofs of the results in this section (and, for k-tuple domination, of the proof by Klasing and Laforest [26]), we obtain the following result.

Theorem 12. *Unless $\mathsf{P} = \mathsf{NP}$, the following holds:*

- *For every problem $\Pi \in \{$domination, k-domination, k-tuple domination, k-tuple total domination, multiple domination, total domination$\}$ (see Table 1) there is no polynomial time algorithm approximating Π within a factor of $0.2267 \ln n$.*
- *For every problem $\Pi \in \{\alpha$-domination, partial monopoly, strict α-domination, vector domination$\}$ (see Table 1) there is no polynomial time algorithm approximating Π within a factor of $0.1133 \ln n$.*
- *For every problem $\Pi \in \{\alpha$-rate domination, monopoly, positive influence domination, strict total α-domination, total α-domination, total vector domination$\}$ (see Table 1) there is no polynomial time algorithm approximating Π within a factor of $0.0755 \ln n$.*

The following corollary is immediate.

Corollary 4. *All the problems listed in Theorem 12 are NP-complete.*

To the best of our knowledge, this is the first NP-completeness proof for α-rate domination, k-tuple total domination, strict α-domination, strict total α-domination, and total α-domination. For references to NP-completeness proofs of the remaining problems, see Table 2.

4. Polynomial algorithms for particular graph classes

In this section, we present several polynomial time algorithms for vector domination and total vector domination in particular graph classes. For notational convenience, we will often replace in this section the vector notation

$\mathbf{k} = (k_v : v \in V)$ with the function notation: an instance to the (total) vector domination problem will be given by a pair (G, k) where $G = (V, E)$ is a graph and $k : V \longrightarrow \mathbb{N}_0$ is a function.

We start with complete graphs.

4.1. Complete graphs

Proposition 1. *Let G be a complete graph with vertex set $V(G) = \{v_1, \ldots, v_n\}$ and assume that $n - 1 \geq k(v_1) \geq \cdots \geq k(v_n) \geq k(v_{n+1}) := 0$ with $k(v_1) > 0$. Then, a minimum vector dominating set for (G, k) is given by $D = \{v_1, \ldots, v_p\}$ where $p = \min\{i : 1 \leq i \leq n, i \geq k(v_{i+1})\}$.*

Proof. Clearly, $D = \{v_1, \ldots, v_p\}$ as above is a vector dominating set for (G, k) since every $v \in V(G) \setminus D$ is of the form $v = v_j$ for some $j \geq p + 1$ and therefore $|N(v_j) \cap D| = |D| = p \geq k(v_{p+1}) \geq k(v_j)$. Conversely, if D is a set of at most $p - 1$ vertices, then there exists a vertex $v_i \in V(G) \setminus D$ such that $i \leq p$. By definition of p, we have $p - 1 < k(v_p)$. Therefore, $|N(v_i) \cap D| \leq p - 1 < k(v_p) \leq k(v_i)$, hence D is not a vector dominating set for (G, k). □

Corollary 5. *For complete graphs, the vector domination problem is solvable in time $O(n)$.*

The claimed time bound can be achieved as follows: Since all the $k(v_i)$'s are less than n, they can be sorted using counting sort in time $O(n)$. Thus, the value of $p = \min\{i : 1 \leq i \leq n, i \geq k(v_{i+1})\}$ and with it a minimum vector dominating set can also be computed within this time bound.

Total vector domination is even simpler and also solvable in $O(n)$ time.

Proposition 2. *Let $G = (V, E)$ be a complete graph. Let $K = \max\{k(v) : v \in V(G)\}$ and let $M = \{v \in V : k(v) = K\}$. If $|M| \leq |V| - K$, then a minimum total vector dominating set for (G, k) is given by any subset of K vertices contained in $V \setminus M$. Otherwise, a minimum total vector dominating set for (G, k) is given by any subset of $K + 1$ vertices.*

Proof. Clearly, every total vector dominating set must contain at least K vertices. Suppose first that $|M| \leq |V| - K$, and let D be a subset of K vertices contained in $V \setminus M$. Then, $k(v) < M$ for every $v \in D$, implying that $|N(v) \cap D| = |D| - 1 = M - 1 \geq k(v)$. For every $v \in V \setminus D$, we have $|N(v) \cap D| = |D| = M \geq k(v)$.

Suppose now that $|M| > |V| - K$. In this case, every subset $D \subseteq V$ with exactly K vertices contains an element from M, say v, therefore $|N(v) \cap D| = |D| - 1 = K - 1 < k(v)$. It follows that every total vector dominating set must contain at least $K+1$ vertices. On the other hand, since G is complete, every set D with $K+1$ vertices will contain at least K neighbors of every vertex. □

Corollary 6. *For complete graphs, the total vector domination problem is solvable in time $O(n)$.*

The claimed time bound can be achieved as follows: Assuming, as usual, that comparing two numbers can be done in constant time, computing the maximum $\max\{k(v) : v \in V(G)\}$ and the set M can be done in time $O(n)$. The rest follows using Proposition 2.

4.2. Trees

Total vector domination and multiple domination problems are solvable in linear time on trees [27] and also in the larger class of strongly chordal graphs [28, 35]. However, in [28] Liao and Chang mention that their approach cannot be modified to handle the case of vector domination, and that a new approach is needed. In this section we describe a linear time algorithm that solves vector domination in trees.

Given a tree T, we root it at an arbitrary vertex r. For a vertex v of T, we denote with T_v the subtree of T rooted at v. For each vertex $v \neq r$ we also use $p(v)$ to denote the parent of v, i.e., the last vertex (v excluded) in the unique path from the root of T to v, and by $C(v)$ the set of children of v.

Theorem 13. *A minimum vector dominating set in a tree can be found in time $O(n)$.*

Proof. We claim that Algorithm 1 computes a minimum vector dominating set for (T, k). The algorithm traverses the tree bottom up, processing vertices one at a time in the opposite order as they are encountered by a breadth-first traversal from the root. If, by the time a node $v_i \neq r$ is processed, at most $k(v_i) - 2$ children of v_i have been put in S (line 7) then there is no other way to satisfy the requirement for S to be a vector dominating set than to add v_i to S, and accordingly the algorithm does so. If $k(v_i) - 1$ children of v_i have been put in S (line 9), then the additional neighbor of v_i to include in S is chosen as v_i's parent $p(v_i)$, for obvious reasons. If $k(v_i)$ (or more) children of v_i have already been put into S, then there is nothing to do. Line 13 takes care of the limit case in which $v_i = r$ and therefore no parent of v_i exists.

For $i \in \{0, 1, \ldots, n\}$, let us denote by P_i and S_i the sets P and S, respectively, after i iterations of the **for** loop. (In particular, $S_0 = P_0 = \emptyset$.) To show the correctness of the algorithm, it suffices to prove the following claim:

Claim: For every $i \in \{0, 1, \ldots, n\}$, the following holds:

Algorithm 1 Vector domination in trees

Input: A tree $T = (V, E)$, a function $k : V \longrightarrow \mathbb{N}_0$.
Output: A set $S \subseteq V$ which is a minimum vector dominating set for (T, k).

1: Fix a root $r \in V(T)$, and let $v_1, \ldots, v_n$ be the vertices of T listed in reverse order with respect to the time they are visited by a breadth-first traversal from r.
2: Set $S = P = \emptyset$.
3: **for all** $i = 1, \ldots, n$ **do**
4:   **if** $v_i \notin P$ **then**
5:     $P \leftarrow P \cup \{v_i\}$
6:     **if** $i < n$ (i.e., v_i is not the root) **then**
7:       **if** $|C(v_i) \cap S| \leq k(v_i) - 2$ **then**
8:         $S \leftarrow S \cup \{v_i\}$
9:       **else if** $|C(v_i) \cap S| = k(v_i) - 1$ **then**
10:         $S \leftarrow S \cup \{p(v_i)\}$
11:         $P \leftarrow P \cup \{p(v_i)\}$
12:       **end if**
13:     **else if** $|C(v_i) \cap S| < k(v_i)$ **then**
14:       $S \leftarrow S \cup \{v_i\}$
15:     **end if**
16:   **end if**
17: **end for**
18: **return** S

(a) For every $v \in P_i \setminus S_i$, it holds that $|N(v) \cap S_i| \geq k(v)$.

(b) There exists a minimum vector dominating set D_i for (T, k) such that $D_i \cap P_i = S_i$.

The correctness of the algorithm follows from part (b) of the claim for $i = n$.

We prove the claim by induction on i. Both statements hold trivially for $i = 0$.

For the inductive step, we consider the two statements separately. First we consider part (a).

Let $i \geq 1$ and suppose that the statement in (a) holds for smaller values of i. If $P_i \setminus S_i = P_{i-1} \setminus S_{i-1}$, the statement holds by the induction hypothesis, since $S_{i-1} \subseteq S_i$. There are only two remaining cases:

(i) $S_i = S_{i-1} \cup \{p(v_i)\}$ and $P_i = P_{i-1} \cup \{v_i, p(v_i)\}$. This corresponds to the case when the **if** statement in line 9 is true. In this case $P_i \setminus S_i = (P_{i-1} \setminus S_{i-1}) \cup \{v_i\}$, and we only need to verify the condition $|N(v_i) \cap S_i| \geq k(v_i)$. This is indeed the case since by the standing hypothesis, exactly $k(v_i) - 1$ of the children of v_i are in S_{i-1}, and S_i contains all of them plus the parent of v_i.

(ii) $S_i = S_{i-1}$ and $P_i = P_{i-1} \cup \{v_i\}$. In this case $P_i \setminus S_i = (P_{i-1} \setminus S_{i-1}) \cup \{v_i\}$, and we only need to verify the condition $|N(v_i) \cap S_i| \geq k(v_i)$. This case only happens when none of the conditions in lines 7, 9 and 13 are satisfied, which means that S_{i-1} contains at least $k(v_i)$ neighbors, in fact, children, of v_i. Thus, the condition holds by the induction hypothesis.

Now we consider part (b). Let $i \geq 1$ and suppose that the statement in (b) holds for smaller values of i.

We may assume that $P_i \neq P_{i-1}$ since otherwise we also have $S_i = S_{i-1}$ and we can take $D_i = D_{i-1}$, applying the inductive hypothesis.

Suppose first that $S_i = S_{i-1}$. Then none of the conditions in lines 7, 9 and 13 are satisfied, which means that S_{i-1} contains at least $k(v_i)$ children of v_i.

If $v_i \notin D_{i-1}$, then we can take $D_i = D_{i-1}$.

If $v_i \in D_{i-1}$, then by the minimality of D_{i-1}, we see that $v_i \neq r$ and $p(v_i) \notin D_{i-1}$, for otherwise we could remove v_i from D_{i-1} to obtain a smaller vector dominating set for (T, k). Let $D_i = (D_{i-1} \setminus \{v_i\}) \cup \{p(v_i)\}$. Clearly, $|D_i| = |D_{i-1}|$ and $D_i \cap P_i = S_i$. Now we argue that D_i is a vector dominating set for (T, k). Notice that $D_{i-1} \setminus \{v_i\}$ vector dominates all vertices of T except possibly $p(v_i)$: for v_i, this is the case due to the fact that S_{i-1} is included in $D_{i-1} \setminus \{v_i\}$ (by the induction hypothesis), and S_{i-1} contains at least $k(v_i)$ children of v_i; moreover, S_{i-1} vector dominates all children of v_i by (a); finally, for every vertex outside $T_{v_i} \cup \{p(v_i)\}$ this is true because it is vector dominated by D_{i-1} and it is not a neighbor of v_i.

Now suppose that $S_i \neq S_{i-1}$.

If $S_i = S_{i-1} \cup \{p(v_i)\}$ then the condition in line 9 is true. Since $C(v_i) \subseteq P_{i-1}$ we have that $C(v_i) \cap D_{i-1} = C(v_i) \cap (P_{i-1} \cap D_{i-1}) = C(v_i) \cap S_{i-1}$ where the last equality follows by the inductive hypothesis. Thus $|C(v_i) \cap D_{i-1}| = |C(v_i) \cap S_{i-1}| = k(v_i) - 1$. Hence, if D_{i-1} does not contain v_i then it must contain $p(v_i)$. If $v_i \notin D_{i-1}$ then it is easy to see that taking $D_i = D_{i-1}$ satisfies the claim. On the other hand, if $v_i \in D_{i-1}$ then we can set $D_i = (D_{i-1} \setminus \{v_i\}) \cup \{p(v_i)\}$. Clearly, $|D_i| \leq |D_{i-1}|$ and $D_i \cap P_i = S_i$. Notice that $D_{i-1} \setminus \{v_i\}$—hence also D_i—vector dominates all vertices of $T_{v_i} - v_i$ by (a) and the induction hypothesis. Also, because of $|D_{i-1} \cap C(v_i)| = k(v_i) - 1$, we have that D_i vector dominates v_i. Finally, $p(v_i)$ is trivially vector dominated by D_i, and so is every remaining vertex v of T because $D_i \cap N(v) \supseteq D_{i-1} \cap N(v)$.

Finally, if $S_i = S_{i-1} \cup \{v_i\}$ then we can take $D_i = D_{i-1}$. It suffices to argue that D_{i-1} contains v_i. Again, since $C(v_i) \subseteq P_{i-1}$ we have that $C(v_i) \cap D_{i-1} = C(v_i) \cap (P_{i-1} \cap D_{i-1}) = C(v_i) \cap S_{i-1}$ where the last equality follows by the inductive hypothesis. By the standing assumption the set $C(v_i) \cap D_{i-1}$ is too small to allow D_{i-1} to vector dominate v_i, unless $v_i \in D_{i-1}$.

This completes the proof of the claim and with it the proof of the correctness of the algorithm.

It remains to analyze the time complexity of the algorithm. A breadth-first traversal together with computing the parents $p(v)$ can be done in linear time. It is not hard to see that all the operations performed by the algorithm at any vertex v_i take constant time: the only operation that requires some care is the computation of the cardinality of set intersection $C(v_i) \cap S$ needed in lines $7, 9, 13$. For this, we keep a counter for each vertex, which is originally set to 0; morevoer, every time we include a new vertex into S, we increase by 1 the counter of its parent (if it exists). Therefore, the algorithm can be implemented to run in linear time. □

4.3. P_4-free graphs

In this section we give a polynomial time algorithm to solve the vector domination and total vector domination problems in P_4-free graphs. P_4-free graphs (also known as cographs) are graphs without an induced subgraph isomorphic to a 4-vertex path. A polynomial time algorithm for k-tuple domination in a class of graphs properly containing the P_4-free graphs was recently given in [9].

In this section, we develop a polynomial time algorithm for the vector domination and total vector domination problems in P_4-free graphs. The algorithm will be based on the following well-known characterization of P_4-free graphs [4]: a graph G is P_4-free if and only if for every induced subgraph F of G with at least two vertices, either F or its complement is disconnected. A *co-component* of a graph $G = (V, E)$ is the subgraph of G induced by the vertex set of a connected component of the complementary graph $\overline{G} = (V, \{uv \mid u, v \in V,\ u \neq v,\ uv \notin E\})$. The above characterization implies that every P_4-free graph $G = (V, E)$ admits a recursive decomposition into one-vertex graphs by taking components or co-components. Such a decomposition can be computed in linear time [5], and a tree representing such a decomposition is called a *cotree*. For our purposes, it will be more convenient to assume that G is represented by a *modified cotree*, which is obtained from the cotree by replacing every node representing a decomposition of an induced subgraph F of G into $p \geq 3$ co-components $C_1, \ldots, C_p$ with $p-1$ nodes in sequence, with i-th node representing the decomposition of $F_i := F - (C_1 \cup \cdots \cup C_{i-1})$ into C_i and $F_i - C_i$.

Proposition 3. *Let G, G_1, G_2 be graphs such that G is obtained from the disjoint union of G_1 and G_2 by adding all edges of the form $\{uv\ : u \in V(G_1),\ v \in V(G_2)\}$. Then,*

$$\gamma(G, k) = \min_{\substack{0 \leq i \leq |V(G_2)| \\ 0 \leq j \leq |V(G_1)|}} \big(\max\{\gamma(G_1, k_i), j\} + \max\{\gamma(G_2, k'_j), i\}\big)$$

$$\gamma^t(G,k) = \min_{\substack{0\le i\le |V(G_2)|\\ 0\le j\le |V(G_1)|}} \left(\max\{\gamma^t(G_1,k_i),j\} + \max\{\gamma^t(G_2,k'_j),i\}\right) ,$$

where $k_i(v) = \max\{k(v)-i,0\}$ *for all* $v \in V(G_1)$ *and* $k'_j(v) = \max\{k(v)-j,0\}$ *for all* $v \in V(G_2)$.

Proof. Let m denote the value of the first minimum above. First, we show that $m \le \gamma(G,k)$. Let D be a minimum vector dominating set for (G,k), that is, $|D| = \gamma(G,k)$. Let $D_i = D \cap V(G_i)$, for $i = 1,2$, and let $i^* = |D_2|$ and $j^* = |D_1|$. Take a vertex $v \in V(G_1) \setminus D_1$ such that $k_{i^*}(v) > 0$. Then

$$|N_{G_1}(v) \cap D_1| = |N_G(v) \cap D| - |D_2| = |N_G(v) \cap D| - i^* \ge k(v) - i^* = k_{i^*}(v)\,.$$

Therefore D_1 is a vector dominating set for (G_1,k_{i^*}) and consequently $\gamma(G_1,k_{i^*}) \le |D_1| = j^*$. Similarly, we can show that $\gamma(G_2,k'_{j^*}) \le |D_2| = i^*$. It follows that

$$\gamma(G,k) = |D| = j^* + i^* = \max\{\gamma(G_1,k_{i^*}),j^*\} + \max\{\gamma(G_2,k'_{j^*}),i^*\} \ge m\,.$$

To see the converse inequality, let (i^*,j^*) be a pair of indices where the value of m is attained. Let D_1 be a vector dominating set for (G_1,k_{i^*}) such that $|D_1| = \max\{\gamma(G_1,k_{i^*}),j^*\}$. Similarly, let D_2 be a vector dominating set for (G_2,k'_{j^*}) such that $|D_2| = \max\{\gamma(G_2,k_{j^*}),i^*\}$. Then, the set $D := D_1 \cup D_2$ is a vector dominating set for (G,k): Let $v \in V(G) \setminus D$. Assuming that $v \in V(G_1) \setminus D_1$, we have

$$|N_G(v) \cap D| = |N_{G_1}(v) \cap D_1| + |D_2| \ge k_{i^*}(v) + |D_2| \ge k(v) - i^* + |D_2| \ge k(v)\,.$$

We can show similarly that $|N_G(v) \cap D| \ge k(v)$ for all $v \in V(G_2) \setminus D_2$. Therefore, $\gamma(G,k) \le |D| = |D_1| + |D_2| = m$, which completes the proof.

The proof of the other relation is analogous. □

Theorem 14. *Vector domination problem and total vector domination problem are solvable in polynomial time on P_4-free graphs.*

Proof. We claim that Algorithm 2 below computes a minimum vector dominating set for (G,k), where G is a P_4-free graph. The following notations are used: For a non-negative integer r and for an induced subgraph H of G, we denote by $D(H,r)$ a minimum vector dominating set for (H,k_r), where $k_r(v) = \max\{k(v)-r,0\}$ for all $v \in V(H)$.

In lines 1–2, the algorithm computes the set R of required vertices in every feasible solution, and reduces the problem to a smaller graph. Notice that once the required vertices have been removed, it holds that $k(v) \le d(v)$ for all v. In

Algorithm 2 Vector domination in P_4-free graphs

Input: A P_4-free graph $G = (V, E)$, a function $k : V \longrightarrow \mathbb{N}_0$.
Output: A minimum vector dominating set for (G, k).

Let $R = \{v \in V(G) \ : k(v) > d(v)\}$.
Set G to $G - R$ and k to $k' : V(G - R) \longrightarrow \mathbb{N}_0$, given by $k'(v) = \max\{k(v) - |N(v) \cap R|, 0\}$ for all $v \in V(G) - R$.
Compute a modified cotree T of G.
for all leaves ℓ of T **do**
  let $v \in V(G)$ be the vertex corresponding to ℓ.
  for all $0 \leq r \leq \Delta(G)$ **do**
    set $D(\{v\}, r) = \begin{cases} \emptyset, & \text{if } k(v) \leq r; \\ \{v\}, & \text{otherwise.} \end{cases}$
  end for
end for
for all internal nodes of T (traversed in a bottom-up manner) **do**
  let H be the subgraph of G corresponding to the current node of T.
  if H is disconnected, with connected components $C_1, \ldots, C_m$ **then**
    for all $0 \leq r \leq \Delta(G)$ **do**
      set $D(H, r) = \cup_{1 \leq i \leq m} D(C_i, r)$.
    end for
  else
    let C be a co-component of H and let $H_2 = H - C$.
    for all $0 \leq r \leq \Delta(G)$ **do**
      for all $0 \leq i \leq |V(H_2)|$ **do**
        let $D_i = D(C, \min\{r + i, \Delta(G)\})$.
      end for
      for all $0 \leq j \leq |V(C)|$ **do**
        let $D'_j = D(H_2, \min\{r + j, \Delta(G)\})$.
      end for
      let (i^*, j^*) be a pair of indices such that $\max\{|D_{i^*}|, j^*\} + \max\{|D'_{j^*}|, i^*\} = \min_{i,j} \left(\max\{|D_i|, j\} + \max\{|D'_j|, i\}\right)$
      let $\hat{D}_1 = D_{i^*} \cup J$ where $J \subseteq V(C) \setminus D_{i^*}$ such that $|J| = \max\{j^* - |D_{i^*}|, 0\}$.
      let $\hat{D}_2 = D'_{j^*} \cup J$ where $J \subseteq V(G_2) \setminus D'_{j^*}$ such that $|J| = \max\{i^* - |D'_{j^*}|, 0\}$.
      set $D(H, r) = \hat{D}_1 \cup \hat{D}_2$.
    end for
  end if
end for
return $D(G, 0) \cup R$.

particular, for an induced subgraph H of the reduced graph $G - R$, it suffices to compute the sets $D(H, r)$ for $r \leq \Delta(G)$, since $D(H, r') = \emptyset$ for every $r' \geq \Delta(G)$.

The correctness of the algorithm is straightforward, using the above-mentioned characterization of P_4-free graphs and Proposition 3 together with the arguments given in its proof. It is also easy to see that the algorithm runs in time $O(\Delta(G)n^3)$.

The algorithm can be modified slightly so that it computes a minimum total vector dominating set. Suppose that an induced subgraph H of G contains a vertex v such that $k(v) - r > d(v)$. In this case, we set $D(H, r) = \text{Inf}$ where Inf is a special symbol denoting the infeasibility of the problem (we also set $|\text{Inf}| = \infty$); moreover Inf is invariant under taking unions: $A \cup \text{Inf} = \text{Inf}$ for every A. We need the following modifications:

- replace lines 1–2 with the following:

  if there exists a vertex v such that $k(v) > d(v)$ **then** return Inf.

- replace line 7 with the following:

  set $D(\{v\}, r) = \begin{cases} \emptyset, & \text{if } k(v) \leq r; \\ \text{Inf}, & \text{otherwise.} \end{cases}$

□

4.4. Threshold graphs

Threshold graphs form a subclass of P_4-free graphs, therefore vector domination and total vector domination problems are solvable in polynomial time on threshold graphs. Since threshold graphs are strongly chordal, the total vector domination problem is solvable in time $O(n + m)$ on threshold graphs [28, 35]. We develop in this section an $O(nm)$ algorithm for the vector domination problem in threshold graphs, using the following characterization: A graph $G = (V, E)$ is threshold if and only if there is an ordering $v_1, \dots, v_n$ of V such that for every i, vertex v_i is either isolated or dominating in the subgraph G_i of G induced by $\{v_1, \dots, v_i\}$. Such an ordering of a threshold graph G can be found in linear time by recursively removing dominating or isolated vertices.

We will also need the following proposition similar to Proposition 3. For a subgraph H of G, we denote by $k|_H$ the restriction of k to $V(H)$, that is, the function $k|_H : V(H) \longrightarrow \mathbb{N}_0$, given by $k|_H(v) = k(v)$ for all $v \in V(H)$.

Proposition 4. *Let G be a graph with a dominating vertex v. Let $G' = G - \{v\}$ and $k' : V(G') \longrightarrow \mathbb{N}_0$ be given by $k'(u) = \max\{k(u) - 1, 0\}$ for all $u \in V(G')$. If $k(v) > d(v)$ then every minimum vector dominating set D for (G, k) is of the form $D' \cup \{v\}$ where D' is a minimum vector dominating set for (G', k'). Otherwise,*

$$\gamma(G, k) = \min\{\max\{\gamma(G', k|_{G'}), k(v)\}, 1 + \gamma(G', k')\} .$$

More specifically, if D' is a minimum vector dominating set for $(G', k|_{G'})$ and D'' is a minimum vector dominating set for (G', k') then a minimum vector dominating set D for (G, k) can be computed as follows:

$$D = \begin{cases} D' \cup J, & \text{if } \max\{|D'|, k(v)\} \leq 1 + \gamma(G', k'); \\ D'' \cup \{v\}, & \text{otherwise,} \end{cases}$$

where $J \subseteq V(G') \setminus D'$ such that $|J| = \max\{k(v) - |D'|, 0\}$.

Proof. If $k(v) > d(v)$ then every minimum vector dominating set D for (G, k) must contain v, and the first statement follows.

Suppose now that $k(v) \leq d(v)$. Let D be a minimum vector dominating set for (G, k). If $v \in D$ then $D' = D \setminus \{v\}$ is a vector dominating set for (G', k'). Therefore, in this case $\gamma(G', k') \leq \gamma(G, k) - 1$ and the inequality $\gamma(G, k) \geq \min\{\max\{\gamma(G', k|_{G'}), k(v)\}, 1 + \gamma(G', k')\}$ follows. If $v \notin D$ then $D' = D \setminus \{v\}$ is a vector dominating set for $(G', k|_{G'})$, moreover $|D'| \geq k(v)$; therefore the inequality $\gamma(G, k) \geq \min\{\max\{\gamma(G', k|_{G'}), k(v)\}, 1 + \gamma(G', k')\}$ holds in this case too.

To see the converse inequality, suppose first that $\max\{\gamma(G', k|_{G'}), k(v)\} \leq 1 + \gamma(G', k')$, and let D' be a minimum vector dominating set for $(G', k|_{G'})$. Let $D = D' \cup J$ where $J \subseteq V(G') \setminus D'$ such that $|J| = \max\{k(v) - |D'|, 0\}$. Then, the set D contains at least $k(v)$ neighbors of v, therefore D is a vector dominating set for (G, k). Similarly, if $\max\{\gamma(G', k|_{G'}), k(v)\} > 1 + \gamma(G', k')$, then letting D'' be a minimum vector dominating set for (G', k'), we can define $D = D'' \cup \{v\}$ to obtain a vector dominating set for (G', k'). In summary, $\gamma(G, k) \leq \min\{\max\{\gamma(G', k|_{G'}), k(v)\}, 1 + \gamma(G', k')\}$; hence equality holds, and the set D is also a minimum vector dominating set for (G, k). □

Proposition 4 leads to Algorithm 3 below for the vector domination problem on threshold graphs.

Theorem 15. *A minimum vector dominating set in a threshold graph can be found in time $O(nm)$.*

Proof. We claim that Algorithm 3 computes a minimum vector dominating set for (G, k), where G is a threshold graph. We use similar notation as in the proof of Theorem 14, except that we denote by $D_{i,j}$ a minimum vector dominating set for (G_i, k_j) where $k_j(v) = \max\{k(v) - j, 0\}$ for all $v \in V(G_i)$. The algorithm will compute, by dynamic programming, all sets $D_{i,j}$, for all $i \in \{1, \dots, n\}$ and all $j \in \{0, 1, \dots, p_i\}$ where p_i is the number of indices $j > i$ such that v_j is dominating in G_j.

The correctness of the algorithm follows by induction on i, using Proposition 4. Notice that for all $i \geq 2$ such that v_i is dominating in G_i, we have

Algorithm 3 Vector domination in threshold graphs

Input: A threshold graph $G = (V, E)$, a function $k : V \longrightarrow \mathbb{N}_0$.
Output: A minimum vector dominating set for (G, k).

1: Let $R = \{v \in V(G) \ : k(v) > d(v)\}$.
2: Set G to $G - R$ and k to $k' : V(G-R) \longrightarrow \mathbb{N}_0$, given by $k'(v) = \max\{k(v) - |N(v) \cap R|, 0\}$ for all $v \in V(G) - R$.
3: Compute an ordering $v_1, \ldots, v_n$ of $V(G)$ such that v_i is either isolated or dominating in G_i.
4: Compute the values p_j for all $j \in \{1, \ldots, n\}$.
5: **for all** $0 \leq j \leq p_1$ **do**
6:   set $D_{1,j} = \begin{cases} \emptyset, & \text{if } k(v_1) \leq j; \\ \{v_1\}, & \text{otherwise.} \end{cases}$
7: **end for**
8: **for all** $i = 2, \ldots, n$ **do**
9:   **if** v_i is isolated in G_i **then**
10:     **for all** $0 \leq j \leq p_i$ **do**
11:       set $D_{i,j} = \begin{cases} D_{i-1,j}, & \text{if } k(v_i) \leq j; \\ D_{i-1,j} \cup \{v_i\}, & \text{otherwise.} \end{cases}$
12:     **end for**
13:   **else**
14:     **for all** $0 \leq j \leq p_i$ **do**
15:       **if** $\max\{|D_{i-1,j}|, k(v) - j\} \leq 1 + |D_{i-1,j+1}|$ **then**
16:         let $J \subseteq V(G_{i-1}) \setminus D_{i-1,j}$ such that $|J| = \max\{k(v) - j - |D_{i-1,j}|, 0\}$.
17:         set $D_{i,j} = D_{i-1,j} \cup J$.
18:       **else**
19:         set $D_{i,j} = D_{i-1,j+1} \cup \{v_i\}$.
20:       **end if**
21:     **end for**
22:   **end if**
23: **end for**
24: **return** $D_{n,0} \cup R$.

$p_{i-1} = p_i + 1$, therefore $j + 1 \leq p_{i-1}$ in lines 15 and 19, so $D_{i-1,j+1}$ has already been computed at that point. The total time complexity is $O(n \sum_{i=1}^{n} p_i) = O(nm)$, and can be improved to $O(n + m)$ if only the minimum size of a vector dominating set is needed. □

5. Concluding remarks

We have studied some algorithmic issues related to natural extensions of the well known concepts of domination and total domination in graphs. We have shown that the problems are approximable to within a logarithmic factor, and proved that this is essentially best possible. We summarize our and other known results in Table 2 (which should be read in conjuction with Table 1). In the last column, we provide a reference for NP-completeness proofs of the corresponding decision problems.

Model	Upper bound	Lower bound	NP-completeness
α-domination	$\ln(2\Delta(G)) + 1$	$(1/2 - \epsilon)\ln n$	[11]
α-rate domination	$\ln(\Delta(G)) + 1$	$(1/3 - \epsilon)\ln n$	Corollary 4
domination	$\ln(\Delta(G) + 1) + 1/2$ [2, 10]	$(1 - \epsilon)\ln n$ [2]	[17]
k-domination	$\ln(2\Delta(G)) + 1$	$(1 - \epsilon)\ln n$	[24]
k-tuple domination	$\ln(\Delta(G) + 1) + 1$ [26]	$(1 - \epsilon)\ln n$ [26]	[28]
k-tuple total domination	$\ln(\Delta(G)) + 1$	$(1 - \epsilon)\ln n$	Corollary 4
monopoly	$\ln(\Delta(G) + 1) + 1$ [33]	$(1/3 - \epsilon)\ln n$ [30]	[30, 33]
multiple domination	$\ln(\Delta(G) + 1) + 1$	$(1 - \epsilon)\ln n$	generalizes domination
partial monopoly	$\ln(2\Delta(G)) + 1$ [33]	$(1/2 - \epsilon)\ln n$ [30]	[30, 33]
positive influence domination	$\ln(\Delta(G)) + 1$ [36]	$(1/3 - \epsilon)\ln n$	[36]
strict α-domination	$\ln(2\Delta(G)) + 1$	$(1/2 - \epsilon)\ln n$	Corollary 4
strict total α-domination	$\ln(\Delta(G)) + 1$	$(1/3 - \epsilon)\ln n$	Corollary 4
total domination	$\ln(\Delta(G)) + 1/2$ [2, 10]	$(1 - \epsilon)\ln n$ [2]	[21]
total α-domination	$\ln(\Delta(G)) + 1$	$(1/3 - \epsilon)\ln n$	Corollary 4
total vector domination	$\ln(\Delta(G)) + 1$	$(1/3 - \epsilon)\ln n$	generalizes total domination
vector domination	$\ln(2\Delta(G)) + 1$	$(1/2 - \epsilon)\ln n$	generalizes domination

Table 2: Known approximability results for different domination problems. Lower bounds hold unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. They also hold unless $\mathsf{P} = \mathsf{NP}$ but the constant must be multiplied by 0.2267. Unless stated otherwise, all the upper and lower bounds in the table are from this paper.

We have also provided exact polynomial time algorithms for several interesting classes of graphs, namely, complete graphs, trees, P_4-free graphs and threshold graphs. We leave it as a question for future research to determine the complexity status of the vector domination and related problems for graphs of bounded tree-width or bounded clique-width.

Acknowledgements

The authors are grateful to Dieter Rautenbach for telling them about the notion of α-domination. The linear time algorithm for vector domination in trees given in Section 4.2 was inspired by discussions with André Nichterlein. The authors are also grateful to two anonymous referees whose comments helped to improve the presentation of the paper.

The first author was partially supported by a DAAD grant, ref. code A/11/15927. The work of the second author was partially done during several visits at the Department of Informatics at the University of Salerno; the kind hospitality of the first and the third authors is greatly appreciated. The second author was supported in part by “Agencija za raziskovalno dejavnost Republike Slovenije”, research program P1-0285 and research projects J1–4010, J1–4021 and N1–0011. The work of the third author was partially done while visiting the Mascotte team of INRIA at Sophia Antipolis. He wants to thank J.-C. Bermond and D. Coudert for their kind hospitality.

References

[1] N. Alon, D. Moshkovitz and S. Safra. Algorithmic construction of sets for k-restrictions. *ACM Transactions on Algorithms* 2 (2006) 153–177.

[2] M. Chlebík and J. Chlebíkova. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation* 206 (2008) 1264–1275.

[3] F. Cicalese, M. Milanič and U. Vaccaro. Hardness, approximability, and exact algorithms for vector domination and total vector domination in graphs. *FCT 2011*, LNCS Vol. 6914 (2011) 288–297.

[4] D.G. Corneil, H. Lerchs and L. Stewart Burlingham. Complement reducible graphs. *Discrete Appl. Math.* 3 (1981) 163–174.

[5] D.G. Corneil, Y. Perl and L.K. Stewart. A linear recognition algorithm for cographs. *SIAM J. Comput.* 14 (1985) 926–934.

[6] F. Dahme, D. Rautenbach and L. Volkmann. Some remarks on α-domination. *Discussiones Mathematicae, Graph Theory* 24 (2004) 423–430.

[7] F. Dahme, D. Rautenbach and L. Volkmann. α-Domination perfect trees. *Discrete Math.* 308 (2008) 3187–3198.

[8] G. Dobson. Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Mathematics of Operations Research* 7 (1982) 515–531.

[9] M.P. Dobson, V. Leoni and G. Nasin. The multiple domination and limited packing problems in graphs. *Information Processing Letters* 111 (2011) 1108–1113.

[10] R. Duh and M. Fürer. Approximation of k-set cover by semi-local optimization. In: Proceedings of the 29th ACM Symposium on Theory of Computing, STOC, 1997, pp. 256–264.

[11] J.E. Dunbar, D.G. Hoffman, R.C. Laskar and L.R. Markus. α-Domination. *Discrete Math.* 211 (2000) 11–26.

[12] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of ACM* 45 (1998) 634–652.

[13] J.F. Fink and M.S. Jacobson. n-domination in graphs. *Graph Theory with Applications to Algorithms and Computer Science.* John Wiley and Sons, New York, 1985, pp. 283–300.

[14] J.F. Fink and M.S. Jacobson. On n-domination, n-dependence and forbidden subgraphs. *Graph Theory with Applications to Algorithms and Computer Science.* John Wiley and Sons, New York, 1985, pp. 301–311.

[15] A. Gagarin, A. Poghosyan and V.E. Zverovich. Upper bounds for α-domination parameters. *Graphs and Combinatorics* 25 (2009) 513–520.

[16] A. Gagarin, A. Poghosyan and V.E. Zverovich. Randomized algorithms and upper bounds for multiple domination in graphs and networks. *Discrete Applied Math.* (2011) doi:10.1016/j.dam.2011.07.004.

[17] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[18] W. Goddard and M.A. Henning. Restricted domination parameters in graphs. *Journal of Combinatorial Optimization* 13 (2007) 353–363.

[19] J. Harant, A. Prochnewski and M. Voigt. On dominating sets and independent sets of graphs. *Combinatorics, Probability and Computing* 8 (1999) 547–553.

[20] F. Harary and T.W. Haynes. Double domination in graphs. *Ars Combin.* 55 (2000) 201–213.

[21] T.W. Haynes, S. Hedetniemi and P. Slater *Fundamentals of Domination in Graphs.* Marcel Dekker, 1998.

[22] T.W. Haynes, S. Hedetniemi and P. Slater (Eds.) *Domination in Graphs: Advanced Topics.* Marcel Dekker, 1998.

[23] M.A. Henning and A.P. Kazemi. k-tuple total domination in graphs. *Discrete Appl. Math.* 158 (2010) 1006–1011.

[24] M. S. Jacobson and K. Peters. *Complexity questions for n-domination and related parameters. Congr. Numer.* 68 (1989) 7–22.

[25] D. Kempe, J.M. Kleinberg and E. Tardos. *Influential Nodes in a Diffusion Model for Social Networks.* ICALP 2005, 1127–1138.

[26] R.Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *Inform. Process. Lett.* 89 (2004) 75–83.

[27] C.S. Liao and G.J. Chang. Algorithmic aspects of k-tuple domination in graphs. *Taiwanese Journal of Mathematics* 6 (2002) 415–420.

[28] C.S. Liao and G.J. Chang. k-tuple domination in graphs. *Inform. Process. Lett.* 87 (2003) 45–50.

[29] S. Mishra. Complexity of majority monopoly and signed domination problems. *J. Discrete Algorithms* 10 (2012) 49–60.

[30] S. Mishra, J. Radhakrishnan and S. Sivasubramanian. On the hardness of approximating minimum monopoly problems. *FST TCS 2002* LNCS Vol. 2556 2002, 277-288

[31] S. Mishra and S. B. Rao. Minimum monopolies in regular and tree graphs. *Discrete Mathematics* 306 (2006) 1586–1594.

[32] E. Mossel, and S. Roch. On the submodularity of influence in social networks. *Proc. 39th Ann. ACM Symp. on Theory of Comp.*, ACM, 2007, pp. 128–134.

[33] D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science* 282 (2002) 231–257.

[34] V. Raman, S. Saurabh and S. Srihari. Parameterized Algorithms for Generalized Domination. *COCOA 2008*, LNCS Vol. 5165 (2008) 116–126.

[35] R. Uehara. Linear time algorithms on chordal bipartite and strongly chordal graphs *Proc. ICALP 2002*, LNCS 2380, 993-1004, 2002.

[36] F. Wang, H. Du, E. Camacho, K. Xu, W. Lee, Y. Shi, and S. Shan On positive influence dominating sets in social networks. *Theoretical Computer Science* 412 (2011) 265–269.

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

[38] F. Zou, J. K Willson, Z. Zhang, and W. Wu. Fast information propagation in social networks. *Discrete Mathematics, Algorithms and Applications* 2 (2010) 1–17.

Appendix A. Proof of Theorem 9

Theorem 9 1. *For every $\alpha \in (0,1)$ and every $\epsilon > 0$, there is no polynomial time algorithm approximating α-rate domination within a factor of $(\frac{1}{3} - \epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

Proof. Let $0 < \alpha < 1$ and $\epsilon \in (0, \frac{1}{3})$. Let $B = \lceil \frac{\alpha}{1-\alpha} \rceil$. We make a reduction from domination on graphs G with n vertices, none of which are isolated, such that

$$n \geq \max\left\{\sqrt{\frac{1-\alpha}{\alpha}}, (B+1)^{2/\epsilon}\right\}, \tag{A.1}$$

$$\lceil n^{1+\epsilon/3} \rceil \leq n^{1+2\epsilon/3}, \tag{A.2}$$

$$n + B \leq n^{1+\epsilon/3}. \tag{A.3}$$

and

$$\gamma(G) \geq \frac{B}{\epsilon}. \tag{A.4}$$

Clearly, these assumptions are without loss of generality since the inequalities in (A.1)–(A.3) are satisfied for all large enough n, while if the inequality (A.4) is violated, we can find an optimal solution in polynomial time by verifying all subsets of $V(G)$ of size less than $\frac{B}{\epsilon}$.

Let G be a graph satisfying $(A.1)$–$(A.4)$. Let $n = |V(G)|$ and $m := \lceil n^{1+\epsilon/3} \rceil$. We transform G into a graph G' as follows: G' consists of mn disjoint copies of G, say $G_1, \ldots, G_{mn}$, together with a complete graph K on Bmn vertices such that K is disjoint from the mn copies of G. (See Fig. 1.) To describe the remaining edges, we first partition the vertex set of K into m equally-sized parts $K_1, \ldots, K_m$. (In particular, $|K_i| = Bn$ for all $i = 1, \ldots, n$.) Finally, for every $j \in \{1, \ldots, mn\}$, we make every vertex $v \in V(G_j)$ adjacent to precisely k_v vertices in $K_{\lceil j/n \rceil}$ where k_v is an integer satisfying

$$\frac{k_v}{d_G(v) + k_v + 1} < \alpha \leq \frac{k_v + 1}{d_G(v) + k_v + 1}.$$

We can take $k_v = \lceil \frac{\alpha d_G(v) + \alpha - 1}{1-\alpha} \rceil$. Notice that since $0 \leq k_v \leq Bn$, it is indeed possible to assign to every $v \in V(G_j)$ precisely k_v neighbors in $K_{\lceil j/n \rceil}$. (This assignment is done in an arbitrary way.)

Claim: $mn\gamma(G) \leq \gamma_{\times\alpha}(G') \leq mn\gamma(G) + Bmn$.

Proof of Claim:

Let S' be an optimal α-rate dominating set in G', that is, $|S'| = \gamma_{\times\alpha}(G')$. For every $j = 1, \ldots, mn$, let $S'_j = S' \cap V(G_j)$ denote the part of S' that belongs to to the j-th copy of G in G'. Pick an index $j^* \in \{1, \ldots, mn\}$ for which the size of S'_j is smallest. We argue that the set $S := S_{j^*}$ is a dominating set in G_{j^*} (and thus in G). Indeed, suppose for contradiction that there exists a vertex v in G_{j^*} such that S misses the closed neighborhood of v. Then $|N_{G'}[v] \cap S'| \leq k_v$ while the size of the closed neighborhood of v in G' is equal to $|N_{G'}[v]| = d_G(v)+1+k_v$. Therefore

$$\frac{|N_{G'}[v] \cap S'|}{|N_{G'}[v]|} \leq \frac{k_v}{d_G(v)+1+k_v} < \alpha\,,$$

contrary to the assumption that S' is α-rate dominating. This implies that $\gamma(G) \leq |S|$ and consequently $mn\gamma(G) \leq mn|S| \leq \sum_{j=1}^{mn} |S'_j| \leq |S'| = \gamma_{\times\alpha}(G')$.

Conversely, let S be an optimal dominating set in G. For $j = 1, \ldots, mn$, let S_j denote the copy of S in G_j, and let $S' = K \cup \bigcup_{j=1}^{mn} S_j$. The set $S' \subseteq V(G')$ satisfies $|S'| = mn\gamma(G) + Bnm$. Moreover, S' is an α-rate dominating set in G':

- For every $j = 1, \ldots, mn$ and for every $v \in V(G_j)$, the set $N_{G'}[v] \cap S'$ is the disjoint union of sets $N_{G_j}[v] \cap S_j$ and $N_{G'}(v) \cap K$. Hence

$$|N_{G'}[v] \cap S'| = |N_{G_j}[v] \cap S_j| + |N_{G'}(v) \cap K| \geq$$

$$\geq 1 + k_v \geq \alpha(d_{G_j}(v) + 1 + k_v) = \alpha|N_{G'}[v]|\,.$$

  The second inequality holds by the choice of k_v.

- Let $v \in K$. By construction of G', v is adjacent to every other vertex in K, and to at most n^2 remaining vertices. Hence $|N_{G'}[v]| \leq |K| + n^2 = Bmn + n^2$. Moreover, $|N_{G'}[v] \cap S'| \geq |K| = Bmn$. Therefore, to show that $|N_{G'}[v] \cap S'| \geq \alpha|N_{G'}[v]|$, it suffices to prove that

$$\frac{Bmn}{Bmn + n^2} \geq \alpha\,.$$

  This is equivalent to $Bm(1-\alpha) \geq \alpha n$, or, equivalently,

$$m \geq \frac{\alpha}{(1-\alpha)B} n\,,$$

  which is true by the definition of m and since $n^{\epsilon/3} \geq 1 \geq \frac{\alpha}{(1-\alpha)B}$.

This shows that $\gamma_{\times\alpha}(G') \leq mn\gamma(G) + Bmn$ and completes the proof of the claim.

Let us write $n' = |V(G')|$. By assumptions (A.2) and (A.3) we have

$$n' = mn^2 + Bmn \leq n^{3+2\epsilon/3} + Bn^{2+2\epsilon/3} = (n+B)n^{2+2\epsilon/3} \leq n^{1+\epsilon/3}n^{2+2\epsilon/3} = n^{3+\epsilon}\,.$$

Suppose that there exists a polynomial time algorithm A' that computes a $(\frac{1}{3} - \epsilon)\ln n'$-approximation to α-rate domination in G'. Let S' be the set computed by A'. Then $|S'| \leq (\frac{1}{3} - \epsilon)(\ln n')\gamma_{\times\alpha}(G')$.

Similarly as in the proof of the claim above, let $S'_j = S' \cap V(G_j)$ and pick an index $j^* \in \{1, \ldots, mn\}$ for which the value of $|S'_j|$ is smallest. Then, setting $S = S'_{j^*}$ results in a dominating set in G_j (and hence in G).

We can bound the size of S from above as follows:

$$
\begin{array}{rcll}
|S| & \leq & \frac{1}{mn} \cdot |S'| & \text{(by the choice of } j^*\text{)} \\
& \leq & \frac{1}{mn}(\frac{1}{3} - \epsilon)(\ln n') \cdot \gamma_{\times\alpha}(G') & \text{(by the assumption on } A'\text{)} \\
& \leq & \frac{1}{mn}(\frac{1}{3} - \epsilon)(\ln(n^{3+\epsilon})) \cdot (mn\gamma(G) + Bnm) & \text{(by the Claim and } n' \leq n^{3+\epsilon}\text{)} \\
& = & (\frac{1}{3} - \epsilon)(3 + \epsilon)(\ln n) \cdot (\gamma(G) + B) & \\
& \leq & (\frac{1}{3} - \epsilon)(3 + \epsilon)(1 + \epsilon)(\ln n) \cdot \gamma(G) & \text{(by (A.4))} \\
& = & (1 - \epsilon')(\ln n)\gamma(G)\,, &
\end{array}
$$

where $\epsilon' = \epsilon^3 + 11\epsilon^2/3 + 5\epsilon/3 \in (0, 1)$. Therefore, S approximates domination within a factor of $(1 - \epsilon')\ln n$. By Theorem 4, this shows that there is no polynomial time algorithm approximating α-rate domination within a factor of $(\frac{1}{3} - \epsilon)\ln n$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. □