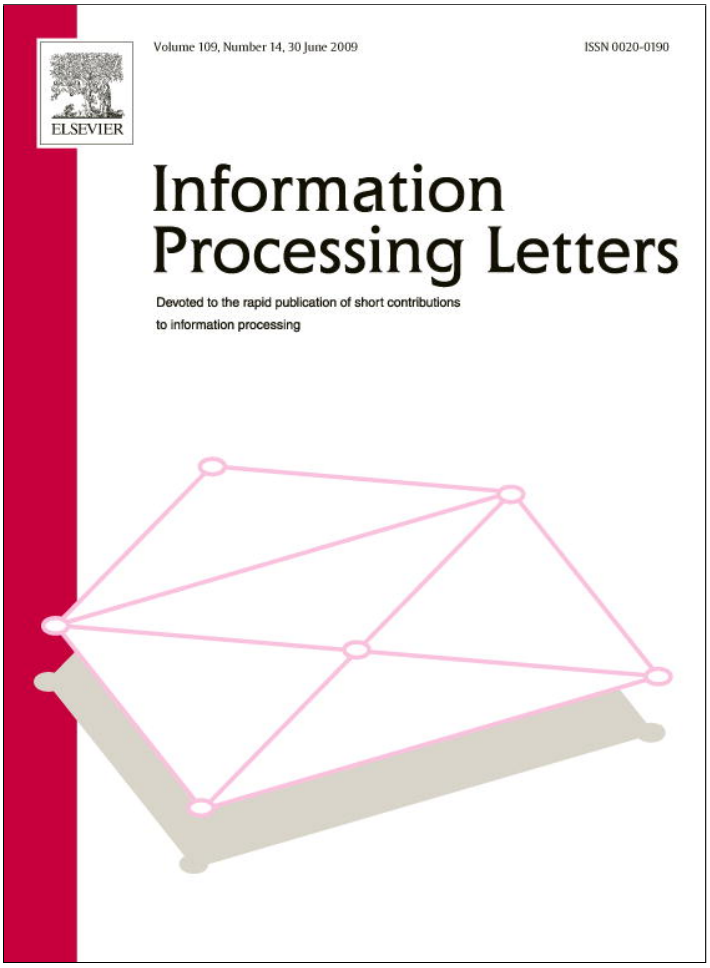
Volume 109, Number 14, 30 June 2009
ISSN 0020-0190
ELSEVIER
Information Processing Letters
Devoted to the rapid publication of short contributions to information processing

Information Processing Letters 109 (2009) 795–798

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Sort and Search: Exact algorithms for generalized domination ☆

Fedor V. Fomin [a,*], Petr A. Golovach [a], Jan Kratochvíl [b], Dieter Kratsch [c], Mathieu Liedloff [d]

[a] Department of Informatics, University of Bergen, 5020 Bergen, Norway
[b] Department of Applied Mathematics, and Institute for Theoretical Computer Science, Charles University, Malostranské nám. 25, 118 00 Praha 1, Czech Republic
[c] Laboratoire d'Informatique Théorique et Appliquée, Université Paul Verlaine - Metz, 57045 Metz Cedex 01, France
[d] Laboratoire d'Informatique Fondamentale d'Orléans, Université d'Orléans, 45067 Orléans Cedex 2, France

ARTICLE INFO

Article history:
Received 19 December 2008
Received in revised form 7 February 2009
Accepted 25 March 2009
Available online 27 March 2009
Communicated by K. Iwama

Keywords:
Exact algorithms
Domination
Graph algorithms

ABSTRACT

In 1994, Telle introduced the following notion of domination, which generalizes many domination-type graph invariants. Let σ and ϱ be two sets of non-negative integers. A vertex subset $S \subseteq V$ of an undirected graph $G = (V, E)$ is called a (σ, ϱ)-dominating set of G if $|N(v) \cap S| \in \sigma$ for all $v \in S$ and $|N(v) \cap S| \in \varrho$ for all $v \in V \setminus S$. In this paper, we prove that decision, optimization, and counting variants of $(\{p\}, \{q\})$-domination are solvable in time $2^{|V|/2} \cdot |V|^{O(1)}$. We also show how to extend these results for infinite $\sigma = \{p + m \cdot \ell\colon\ \ell \in \mathbb{N}_0\}$ and $\varrho = \{q + m \cdot \ell\colon\ \ell \in \mathbb{N}_0\}$. For the case $|\sigma| + |\varrho| = 3$, these problems can be solved in time $3^{|V|/2} \cdot |V|^{O(1)}$, and similarly to the case $|\sigma| = |\varrho| = 1$ it is possible to extend the algorithm for some infinite sets.

1. Introduction

Let $G = (V, E)$ be a finite undirected graph without loops or multiple edges. Here V is the set of vertices and E the set of edges. Throughout the paper we reserve $n = |V|$. We call two vertices u, v *adjacent* if they form an edge, i.e., if $uv \in E$. The *open neighborhood* of a vertex $u \in V$ is the set of the vertices adjacent to it, denoted by $N(u) = \{x\colon\ xu \in E\}$. A set of vertices $S \subseteq V$ is *dominating* if every vertex of G is either in S or adjacent to a vertex in S. Finding a dominating set of the smallest possible size is one of the basic optimization problems on graphs. This problem is also known to be notoriously hard. The problem is NP-hard even for chordal graphs (cf. [6]), and the parameterized version is W[2]-complete [2].

Many generalizations have been studied, such as independent dominating set, connected dominating set, efficient dominating set, etc. (cf. [6]). In [10], Telle introduced the following framework of domination-type graph invariants. Let σ and ϱ be two non-empty sets of non-negative integers. A vertex subset $S \subseteq V$ of an undirected graph $G = (V, E)$ is called a (σ, ϱ)*-dominating set* of G if $|N(v) \cap S| \in \sigma$ for all $v \in S$ and $|N(v) \cap S| \in \varrho$ for all $v \in V \setminus S$. Table 1 shows a sample of previously defined and studied graph invariants which can be expressed in this framework.

We are interested in the computational complexity of decision, search and counting problems related to (σ, ϱ)-domination. Explicitly, we consider the following problems for some special sets σ and ϱ.

$\exists(\sigma, \varrho)$-DS: Does an input graph G contain a (σ, ϱ)-dominating set?

#-(σ, ϱ)-DS: Given a graph G, determine the number of (σ, ϱ)-dominating sets of G.

Max-(σ, ϱ)-DS: Given a graph G, find a (σ, ϱ)-dominating set of maximum size.

Min-(σ, ϱ)-DS: Given a graph G, find a (σ, ϱ)-dominating set of minimum size.

☆ A preliminary version of these results appeared in proceedings of WADS'07 [F.V. Fomin, P. Golovach, D. Kratsch, J. Kratochvíl, M. Liedloff, Branch and recharge: Exact algorithms for generalized domination, in: Proceedings of WADS 2007, in: LNCS, vol. 4619, Springer, 2007, pp. 508–519].

* Corresponding author.
E-mail addresses: fomin@ii.uib.no (F.V. Fomin), petrg@ii.uib.no (P.A. Golovach), honza@kam.ms.mff.cuni.cz (J. Kratochvíl), kratsch@univ-metz.fr (D. Kratsch), liedloff@univ-orleans.fr (M. Liedloff).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2009.03.023

Table 1
Examples of (σ,ϱ)-dominating sets, $\mathbb{N}$ is the set of positive integers, $\mathbb{N}_0$ is the set of non-negative integers.

σ	ϱ	(σ,ϱ)-dominating set
$\mathbb{N}_0$	$\mathbb{N}$	dominating set
$\{0\}$	$\mathbb{N}_0$	independent set
$\mathbb{N}_0$	$\{1\}$	efficient dominating set
$\{0\}$	$\{1\}$	1-perfect code
$\{0\}$	$\{0,1\}$	strong stable set
$\{0\}$	$\mathbb{N}$	independent dominating set
$\{1\}$	$\{1\}$	total perfect dominating set
$\mathbb{N}$	$\mathbb{N}$	total dominating set
$\{1\}$	$\mathbb{N}_0$	induced matching
$\{r\}$	$\mathbb{N}_0$	r-regular induced subgraph

It is interesting to note that already the existence problem is NP-complete for many parameter pairs σ and ϱ, including some of those listed in Table 1 (1-perfect code and total perfect dominating set). In fact, Telle [10] proves that $\exists(\sigma,\varrho)$-DS is NP-complete for every two finite nonempty sets σ,ϱ such that $0\notin\varrho$.

In this paper we show that for a sufficiently large set of decision, optimization, and even counting (σ,ϱ)-dominating problems there are exact algorithms of running time $O^*(2^{n/2})$.[1] Our approach is built on a classical technique of Horowitz and Sahni [7], and Schroeppel and Shamir [9] (see also the survey of Woeginger [11]). The basic idea is a clever use of sorting and searching, and thus we call it Sort and Search.

Let us briefly recall the main ideas of this paradigm. The original problem of size n, say an input graph G on n vertices, is divided into two subproblems, say two disjoint vertex subsets V_1 and V_2 of size $n/2$. For each subset $S\subseteq V_i$ $(i\in\{1,2\})$ a vector of length n is assigned and stored in a table T_i. The definition of the vectors is of course problem dependent. Now T_1 and T_2 contain each at most $2^{n/2}$ different vectors. Then each solution of the problem corresponds to a vector $\vec{a}$ of the first subproblem and a vector $\vec{b}$ of the second one such that the sum of the two vectors is a fixed goal vector $\vec{c}$. All such pairs $(\vec{a},\vec{b})$ of satisfying vectors can be found by searching for each first vector $\vec{a}\in T_1$ the vector $\vec{c}-\vec{a}$ in T_2. When the vectors of the second table are sorted in lexicographic order in a preprocessing, then searching a vector can be done in $O(n)$ times the length of the vectors, and thus the overall running time of the algorithm is $O^*(2^{n/2})$. For more details on searching in a lexicographically ordered table, we refer to vol. 3 of "The Art of Computer Programming" by Knuth [8, p. 409 ff.].

We establish $O^*(2^{n/2})$ time algorithms for the $\exists(\sigma,\varrho)$-DS, Min-(σ,ϱ)-DS, Max-(σ,ϱ)-DS and the #-(σ,ϱ)-DS problem in the case that σ and ϱ are singletons. These results are extended to infinite $\sigma=\{p+m\cdot\ell\colon\ \ell\in\mathbb{N}_0\}$ and $\varrho=\{q+m\cdot\ell\colon\ \ell\in\mathbb{N}_0\}$, for $m\geqslant 2$ and $p,q\in\{0,1,\dots,m-1\}$. Finally, we show that for the case $|\sigma|+|\varrho|=3$, these problems can be solved in time $O^*(3^{n/2})$, and similarly to the case $|\sigma|=|\varrho|=1$ it is possible to generalize the algorithm for some infinite sets.

[1] As has recently become standard, we write $f(n)=O^*(g(n))$ if $f(n)\leqslant p(n)\cdot g(n)$ for some polynomial $p(n)$.

2. Sort and Search algorithms for the case $|\sigma|=|\varrho|=1$

Even very special case of $\exists(\sigma,\varrho)$-DS, namely Perfect Code ($\exists(\{0\},\{1\})$-DS), is NP-complete. It is known that Perfect Code can be solved in time $O(1.1730^n)$ by reduction to the exact satisfiability problem (called XSAT) [1]. Our use of Sort and Search is inspired by the aforementioned algorithms.

Theorem 1. *$\exists(\{p\},\{q\})$-DS, #-$(\{p\},\{q\})$-DS, Max-$(\{p\},\{q\})$-DS and Min-$(\{p\},\{q\})$-DS are solvable in time $O^*(2^{n/2})$.*

Proof. Let $p,q\in\mathbb{N}_0$. Let $G=(V,E)$ be the input graph and let $k=\lfloor n/2\rfloor$. As explained in the introduction, the algorithm partitions the set of vertices into $V_1=\{v_1,v_2,\dots,v_k\}$ and $V_2=\{v_{k+1},\dots,v_n\}$. Then for each subset $S_1\subseteq V_1$, it computes the vector $\vec{s_1}=(x_1,\dots,x_k,x_{k+1},\dots,x_n)$ where

$$x_i=\begin{cases} p-|N(v_i)\cap S_1| & \text{if } 1\leqslant i\leqslant k \text{ and } v_i\in S_1,\\ q-|N(v_i)\cap S_1| & \text{if } 1\leqslant i\leqslant k \text{ and } v_i\notin S_1,\\ |N(v_i)\cap S_1| & \text{if } k+1\leqslant i\leqslant n,\end{cases}$$

and for each subset $S_2\subseteq V_2$, it computes the corresponding vector $\vec{s_2}=(x_1,\dots,x_k,x_{k+1},\dots,x_n)$ where

$$x_i=\begin{cases} |N(v_i)\cap S_2| & \text{if } 1\leqslant i\leqslant k,\\ p-|N(v_i)\cap S_2| & \text{if } k+1\leqslant i\leqslant n \text{ and } v_i\in S_2,\\ q-|N(v_i)\cap S_2| & \text{if } k+1\leqslant i\leqslant n \text{ and } v_i\notin S_2.\end{cases}$$

After computing all these vectors (the total number of vectors is at most 2^{k+1}), we sort vectors corresponding to V_2 lexicographically. Then for each vector $\vec{s_1}$ representing $S_1\subseteq V_1$, we use binary search to tests whether there exists a vector $\vec{s_2}$ representing $S_2\subseteq V_2$, such that $\vec{s_2}=\vec{s_1}$. Note that the choice of the vectors guarantees that $\vec{s_2}=\vec{s_1}$ if and only if $S_1\cup S_2$ is a $(\{p\},\{q\})$-dominating set. Such a vector $\vec{s_1}$ can be found in time $n\log 2^{n/2}$ among the lexicographically ordered vectors of V_2. Thus $\exists(\{p\},\{q\})$-DS is solvable in time $O^*(2^{n/2})$. the overall running time is $O^*(2^{n/2})$.

Now we consider #-$(\{p\},\{q\})$-DS. The algorithm of the previous theorem only needs to be modified as follows: Instead of storing all vectors corresponding to V_1 and V_2 multiple copies are removed and each vector is stored with an entry indicating its number of occurrences. Denote by X_1 the set of all different vectors corresponding to subsets of V_1, and by X_2 the set of vectors corresponding to subsets of V_2. Let $\#_1(\vec{s_1})$ be the number of subsets of V_1 which correspond to $\vec{s_1}\in X_1$, and let $\#_2(\vec{s_2})$ be the number of subsets of V_2 corresponding to $\vec{s_2}$. As for $\exists(\{p\},\{q\})$-DS, for every $\vec{s}\in X_1$, we check whether $\vec{s}$ is included to X_2 as well. Then the number of different (σ,ϱ)-dominating sets is

$$\sum_{\vec{s}\in X_1\cap X_2}\#_1(\vec{s})\cdot\#_2(\vec{s})$$

if $X_1\cap X_2\neq\emptyset$, and this number is 0 otherwise.

Furthermore, for Max-$(\{p\},\{q\})$-DS, with each vector $\vec{s}\in X_i$ we store the subset $S_i(\vec{s})\subseteq V_i$ of maximum cardinality that generates this vector. It can be easily seen that a (σ,ϱ)-dominating set of maximum size (if it exists) is the set $S = S_1(\vec{s^*}) \cup S_2(\vec{s^*})$ such that $\vec{s^*}$ is a vector of $X_1 \cap X_2$ with $|S_1(\vec{s^*})| + |S_2(\vec{s^*})| = \max_{\vec{s} \in X_1 \cap X_2} |S_1(\vec{s})| + |S_2(\vec{s})|$. It is not hard to see that Min-$(\{p\}, \{q\})$-DS can be solved in the same way by replacing *maximum* by *minimum*. □

Now we extend our approach to certain infinite σ and ϱ. Let $m \geqslant 2$ be a fixed integer and $k \in \{0, 1, \ldots, m-1\}$. We denote by $k + m\mathbb{N}_0$ the set $\{k + m \cdot \ell\colon\ \ell \in \mathbb{N}_0\}$.

Theorem 2. *Let $m \geqslant 2$ and $p, q \in \mathbb{N}_0$. The problems $\exists(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS, #-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS, Max-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS, and Min-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS are solvable in time $O^*(2^{n/2})$.*

Proof. For #-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS, the algorithm in Theorem 1 is modified such that for each subset $S_1 \subseteq V_1$, we compute the vector $\vec{s_1} = (x_1, \ldots, x_k, x_{k+1}, \ldots, x_n)$ where

$$x_i = \begin{cases} (p - |N(v_i) \cap S_1|) \bmod m & \text{if } 1 \leqslant i \leqslant k \text{ and } v_i \in S_1, \\ (q - |N(v_i) \cap S_1|) \bmod m & \text{if } 1 \leqslant i \leqslant k \text{ and } v_i \notin S_1, \\ |N(v_i) \cap S_1| \bmod m & \text{if } k+1 \leqslant i \leqslant n, \end{cases}$$

and for each subset $S_2 \subseteq V_2$, the algorithm computes the corresponding vector $\vec{s_2} = (x_1, \ldots, x_k,\ x_{k+1}, \ldots, x_n)$, where

$$x_i = \begin{cases} |N(v_i) \cap S_2| \bmod m & \text{if } 1 \leqslant i \leqslant k, \\ (p - |N(v_i) \cap S_2|) \bmod m & \\ \qquad \text{if } k+1 \leqslant i \leqslant n \text{ and } v_i \in S_2, \\ (q - |N(v_i) \cap S_2|) \bmod m & \\ \qquad \text{if } k+1 \leqslant i \leqslant n \text{ and } v_i \notin S_2. \end{cases}$$

Again, after computing at most 2^{k+1} vectors, the algorithm sorts the vectors representing V_2 lexicographically. By making use of binary search, for each vector $\vec{s_1}$ representing $S_1 \subseteq V_1$, we search for a vector $\vec{s_2}$, representing some $S_2 \subseteq V_2$, and such that $\vec{s_2} = \vec{s_1}$.

For #-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS, Max-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS and Min-$(p + m\mathbb{N}_0, q + m\mathbb{N}_0)$-DS, the modification of the algorithm is similar to the one from Theorem 1, and we omit it here. □

The results of Theorem 2 can be used for the case when σ and ϱ are the sets of even or odd integers [3,5]. These problems are of importance in the coding theory. Let EVEN be the set of all even non-negative integers and ODD be the set of odd positive integers. It was shown in [5] that $\exists$(EVEN, EVEN)-DS, $\exists$(EVEN, ODD)-DS, $\exists$(ODD, EVEN)-DS and $\exists$(ODD, ODD)-DS can be solved in polynomial time while maximization and minimization problems are NP-hard. The next claim follows immediately from Theorem 2.

Corollary 3. *For $\sigma, \varrho \in \{\text{EVEN}, \text{ODD}\}$, the problems #-$(\sigma, \varrho)$-DS, Max-$(\sigma, \varrho)$-DS and Min-$(\sigma, \varrho)$-DS are solvable in time $O^*(2^{n/2})$.*

Variants of these problems for red/blue bipartite graphs were considered in [3]. Suppose that $G = (R, B, E)$ is a bipartite graph with R, B a bipartition of the vertex set. Vertices of R are called *red* and vertices of B are *blue*. Let $S \subseteq R$ be a non-empty set of red vertices. It is said that S is an *even* set if for every vertex $v \in B$, $|N(v)| \in$ EVEN, and S is an *odd* set if for every vertex $v \in B$, $|N(v)| \in$ ODD. The proof of the following theorem is based on combining the Sort and Search approach with dynamic programming.

Theorem 4. *Let $G = (R, B, E)$ be a red/blue bipartite graph. All even or odd sets can be counted, and maximum or minimum even or odd sets can be found in time $O^*(2^{\min\{|R|/2, |B|\}}) = O^*(2^{n/3})$.*

Proof. We prove this claim for the counting problem for even sets. (All other problems can be solved similarly.) Let $R = \{u_1, \ldots, u_k\}$ and $B = \{v_1, \ldots, v_r\}$.

If $k/2 \leqslant r$, then we apply the following Sort and Search algorithm. Let $s = \lfloor k/2 \rfloor$. We partition the set of vertices R into $R_1 = \{u_1, \ldots, u_s\}$ and $R_2 = \{u_{s+1}, \ldots, u_k\}$. For each subset $S_1 \subseteq R_1$, we compute its corresponding vector $\vec{s_1} = (x_1, \ldots, x_r)$, where

$$x_i = \begin{cases} 0 & \text{if } |N(v_i) \cap R_1| \in \text{EVEN}, \\ 1 & \text{if } |N(v_i) \cap R_1| \in \text{ODD}. \end{cases}$$

Similarly for each subset $S_2 \subseteq R_2$, we compute the corresponding vector $\vec{s_2} = (x_1, \ldots, x_r)$, such that

$$x_i = \begin{cases} 0 & \text{if } |N(v_i) \cap R_2| \in \text{EVEN}, \\ 1 & \text{if } |N(v_i) \cap R_2| \in \text{ODD}. \end{cases}$$

Denote by X_1 the set of all different vectors corresponding to subsets of V_1, and by X_2 the set of vectors corresponding to subsets of V_2. Let $\#_1(\vec{s_1})$ be the number of subsets of R_1 corresponding to $\vec{s_1} \in X_1$, and let $\#_2(\vec{s_2})$ be the number of subsets of R_2 corresponding to $\vec{s_2}$. After vectors are computed, we sort the vectors of X_2 lexicographically. For each vector $\vec{s_1} \in X_1$ we search for a vector $\vec{s_2} \in X_2$ such that $\vec{s_2} = \vec{s_1}$. The total number of non-empty even sets is

$$\sum_{\vec{s} \in X_1 \cap X_2} \#_1(\vec{s}) \cdot \#_2(\vec{s}) - 1,$$

and then the running time of this procedure is $O^*(2^{|R|/2})$.

For the case $k/2 > r$, we use dynamic programming approach across the subsets. For every subset $S \subseteq R$, let $\vec{s}(S) = (x_1, \ldots, x_r)$, where

$$x_i = \begin{cases} 0 & \text{if } |N(v_i) \cap S| \in \text{EVEN}, \\ 1 & \text{if } |N(v_i) \cap S| \in \text{ODD}. \end{cases}$$

For every $i \in \{1, \ldots, k\}$, and every vector $\vec{s} \in \mathbb{Z}_2^r$, we put

$$\#(i, \vec{s}) = \left|\left\{S \subseteq \{u_1, \ldots, u_i\}\colon\ \vec{s}(S) = \vec{s}\right\}\right|.$$

We also put $\#(0, \vec{s}) = 0$ for all non-zero vectors $\vec{s}$, and $\#(0, \vec{0}) = 1$. For $i \in \{1, \ldots, k\}$, we denote by $\vec{z_i}$ the vector $(y_1, \ldots, y_r)$, where

$$y_j = \begin{cases} 1 & \text{if } v_j \in N(u_i), \\ 0 & \text{if } v_j \notin N(u_i). \end{cases}$$

Since $\#(i, \vec{s}) = \#(i-1, \vec{s}) + \#(i-1, \vec{s} + \vec{z_i})$, we have that all values $\#(i, \vec{s})$ can be computed in time $O^*(2^{|B|})$ by a dynamic programming approach considering the values i by increasing order. It remains to note that the number of non-empty even sets is $\#(k, \vec{0}) - 1$. □

3. Extending the Sort and Search approach

It is possible to extend (albeit with a worse running time) our results for single-element sets for the case when one set contains two elements and the other set is a singleton.

Theorem 5. *The problems* $\exists(\sigma,\varrho)$-DS, #-(σ,ϱ)-DS, Max-(σ,ϱ)-DS, *and* Min-(σ,ϱ)-DS *are solvable in time* $O^*(3^{n/2})$ *if* $|\sigma|+|\varrho|=3$.

Proof. We prove the theorem for $\exists(\sigma,\varrho)$-DS and $\sigma=\{p_1,p_2\}$, $\varrho=\{q\}$. Let $G=(V,E)$ be a graph and $k=\lfloor n/2\rfloor$. As in all algorithms above, we partition the set of vertices V into $V_1=\{v_1,v_2,\ldots,v_k\}$ and $V_2=\{v_{k+1},\ldots,v_n\}$. Now for every partition of V_1 into three sets $\{S_1^{(1)},S_2^{(1)},\overline{S}^{(1)}\}$ (some of these sets can be empty), we compute the vector $\vec{s_1}=(x_1,\ldots,x_k,x_{k+1},\ldots,x_n)$ where

$$x_i=\begin{cases} p_1-|N(v_i)\cap S_1^{(1)}| & \text{if } 1\leqslant i\leqslant k \text{ and } v_i\in S_1^{(1)},\\ p_2-|N(v_i)\cap S_2^{(1)}| & \text{if } 1\leqslant i\leqslant k \text{ and } v_i\in S_2^{(1)},\\ q-|N(v_i)\cap \overline{S}^{(1)}| & \text{if } 1\leqslant i\leqslant k \text{ and } v_i\in \overline{S}^{(1)},\\ |N(v_i)\cap(S_1^{(1)}\cup S_2^{(1)})| & \text{if } k+1\leqslant i\leqslant n. \end{cases}$$

Symmetrically, for each partition of V_2 into three sets $\{S_1^{(2)},S_2^{(2)},\overline{S}^{(2)}\}$, we compute the corresponding vector $\vec{s_2}=(x_1,\ldots,x_k,\ x_{k+1},\ldots,x_n)$, where

$$x_i=\begin{cases} |N(v_i)\cap(S_1^{(2)}\cup S_2^{(2)})| \\ \quad \text{if } 1\leqslant i\leqslant k,\\ p_1-|N(v_i)\cap S_1^{(2)}| \\ \quad \text{if } k+1\leqslant i\leqslant n \text{ and } v_i\in S_1^{(2)},\\ p_2-|N(v_i)\cap S_2^{(2)}| \\ \quad \text{if } k+1\leqslant i\leqslant n \text{ and } v_i\in S_2^{(2)},\\ q-|N(v_i)\cap \overline{S}^{(2)}| \\ \quad \text{if } k+1\leqslant i\leqslant n \text{ and } v_i\in \overline{S}^{(2)}. \end{cases}$$

After computing these 3^{k+1} vectors, the algorithm sorts vectors of V_2 lexicographically, and for each vector $\vec{s_1}$ (corresponding to a partition of V_1), search for a vector $\vec{s_2}$ from V_2, such that $\vec{s_2}=\vec{s_1}$. Note that $\vec{s_2}=\vec{s_1}$ if and only if $(S_1^{(1)}\cup S_2^{(1)})\cup(S_1^{(2)}\cup S_2^{(2)})$ is a $(\{\sigma\},\{\varrho\})$-dominating set. Since the search of $\vec{s_2}$ can be done in time $n\log 3^{n/2}$, we have that the overall running time of the algorithm is $O^*(3^{n/2})$.

The problems $\exists(\sigma,\varrho)$-DS with $\sigma=\{p\}$ and $\varrho=\{q_1,q_2\}$ are solved similarly. Moreover the algorithm can easily be extended to solve the counting, maximization and minimization version of the problem as it was done in Theorem 1 for single-element sets. □

The algorithms of Theorem 5 can be modified to handle some infinite sets as it was done in Theorem 2. In that case, all components of vectors are taken modulo m and the addition and/or subtraction of vector components is taken modulo m.

Corollary 6. *Let* $m\geqslant 2$ *and* $p_1,p_2,q_1,q_2\in\mathbb{N}_0$. *The problems* $\exists(\sigma,\varrho)$-DS, #-(σ,ϱ)-DS, Max-(σ,ϱ)-DS *and* Min-(σ,ϱ)-DS *are solvable in time* $O^*(3^{n/2})$ *for pairs of sets* $\sigma=(p_1+m\mathbb{N}_0)\cup(p_2+m\mathbb{N}_0)$, $\varrho=q_1+m\mathbb{N}_0$ *and* $\sigma=p_1+m\mathbb{N}_0$, $\varrho=(q_1+m\mathbb{N}_0)\cup(q_2+m\mathbb{N}_0)$.

4. Conclusion

We considered exact algorithms for (σ,ϱ)-dominating set problems for some special sets σ and ϱ, assuming they are the same for all vertices. However it is possible to define a more general problem. Let G be a graph such that for any vertex $v\in V$, two non-empty sets of non-negative integers $\sigma(v)$ and $\rho(v)$ are given. A vertex subset $S\subseteq V$ of the graph G is called now a (σ,ϱ)*-dominating set* of G if $|N(v)\cap S|\in\sigma(v)$ for all $v\in S$ and $|N(v)\cap S|\in\varrho(v)$ for all $v\in V\setminus S$. It should be noted that all of our algorithms can be adopted to solve these problems too.

A natural open question is whether (σ,ϱ)-dominating set problem can be solved in time $(2-\varepsilon)^n$ for some $\varepsilon>0$ for any choice of sets σ and ϱ. It does not seem that Sort and Search can be used to settle this question. In [4], we suggested a different approach for obtaining $(2-\varepsilon)^n$ algorithms for various choices of σ and ϱ, but we are still far from the complete answer.

References

[1] V. Dahlöf, P. Jonsson, R. Beigel, Algorithms for four variants of the exact satisfiability problem, Theoret. Comput. Sci. 320 (2004) 373–394.

[2] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer-Verlag, New York, 1999.

[3] R.G. Downey, M.R. Fellows, A. Vardi, G. Whitte, The parameterized complexity of some fundamental problems in coding theory, SIAM J. Comput. 29 (1999) 545–570.

[4] F.V. Fomin, P. Golovach, D. Kratsch, J. Kratochvíl, M. Liedloff, Branch and recharge: Exact algorithms for generalized domination, in: Proceedings of WADS 2007, in: LNCS, vol. 4619, Springer, 2007, pp. 508–519.

[5] M.M. Halldorsson, J. Kratochvíl, J.A. Telle, Mod-2 independence and domination in graphs, Internat. J. Found. Comput. Sci. 11 (2000) 355–363.

[6] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.

[7] E. Horowitz, S. Sahni, Computing partitions with applications to the knapsack problem, J. ACM 21 (1974) 277–292.

[8] D.E. Knuth, Sorting and Searching, second edition, The Art of Computer Programming, vol. 3, Addison-Wesley, 1997.

[9] R. Schroeppel, A. Shamir, A $T=O(2^{n/2})$, $S=O(2^{n/4})$ algorithm for certain NP-complete problems, SIAM J. Comput. 3 (1981) 456–464.

[10] J.A. Telle, Complexity of domination-type problems in graphs, Nordic J. Comput. 1 (1994) 157–171.

[11] G.J. Woeginger, Exact algorithms for NP-hard problems: A survey, in: Combinatorial Optimization – Eureka, You Shrink!, in: LNCS, vol. 2570, Springer-Verlag, Berlin, 2003, pp. 185–207.