Combinatorial Bounds via Measure and Conquer: Bounding Minimal Dominating Sets and Applications *

Fedor V. Fomin†
Department of Informatics
University of Bergen
PO Box 7800, 5020 Bergen, Norway
fomin@ii.uib.no

Fabrizio Grandoni‡
Dipartimento di Informatica, Sistemi e Produzione
Università di Roma "Tor Vergata"
via del Politecnico 1, 00133 Roma, Italy
grandoni@disp.uniroma2.it

Artem V. Pyatkin§
Sobolev Institute of Mathematics
Siberian Branch of RAS
630090, pr-t Koptyuga, 4, Novosibirsk, Russia
artem@math.nsc.ru

Alexey A. Stepanov
Department of Informatics
University of Bergen
PO Box 7800, 5020 Bergen, Norway
ljosha@ljosha.org

May 8, 2008

Abstract

We provide an algorithm listing all minimal dominating sets of a graph on n vertices in time $\mathcal{O}(1.7159^n)$. This result can be seen as an algorithmic proof of the fact that the number of minimal dominating sets in a graph on n vertices is at most 1.7159^n, thus improving on the trivial $\mathcal{O}(2^n/\sqrt{n})$ bound. Our result makes use of the measure and conquer technique which was recently developed in the area of exact algorithms.

Based on this result, we derive an $\mathcal{O}(2.8718^n)$ algorithm for the domatic number problem.

Keywords: *exact exponential algorithms, listing algorithms, measure and conquer, minimum dominating set, minimum set cover, domatic number*

1 Introduction

One of the typical questions in graph theory is: how many subgraphs satisfying a given property can a graph on n vertices contain? For example, the number of perfect matchings in a simple k-regular bipartite graph on $2n$ vertices is always between $n!(k/n)^n$ and $(k!)^{n/k}$. (The first inequality was known as van der Waerden Conjecture [31] and was proved in 1980 by

*A preliminary version of this paper appeared in ISAAC'05 [15]

†Additional support by the Research Council of Norway.

‡Supported by project MAINSTREAM of the Italian Ministry of University and Research (MIUR).

§Additional support by grants of the Russian Foundation for Basic Research (project code 08-01-00516) and INTAS (project code 04–77–7173)

Egorychev [9] and the second is due to Bregman [5].) Another example is the famous Moon and Moser [25] theorem stating that every graph on n vertices has at most $3^{n/3}$ maximal cliques (independent sets).

The dominating set problem is a classic NP-complete graph optimization problem which fits into the broader class of domination and covering problems. Hundreds of papers have been written on them (see e.g. the survey [20] by Haynes et al.). However, despite the importance of minimum dominating set problem, nothing better than the trivial $\mathcal{O}(2^n/\sqrt{n})$ bound (which is roughly the maximum number of subsets of an n-set such that none of them is contained in the other) was known on the number of minimal dominating sets in a graph.

Our interest is motivated by the design of fast exponential-time algorithms for hard problems. The story of such algorithms dates back to the sixties and seventies. In 1962 Held and Karp presented an $\mathcal{O}(2^n n^2)$ time algorithm for the travelling salesman problem which is still the fastest one known [21]. In 1977 Tarjan and Trojanowski [30] gave an $\mathcal{O}(2^{n/3})$ algorithm for the maximum independent set problem. The last decade has seen a growing interest in fast exponential-time algorithms for NP-hard problems. Examples of recently developed fast exponential algorithms are algorithms for maximum independent set [1, 12], satisfiability [8, 22, 32], coloring [10], treewidth [16], and many others. For a good overview of the field we refer to the recent survey written by Woeginger [33].

It appears that combinatorial bounds are of interests not only on their own but also because they are used for algorithm design as well. Lawler [24] used Moon-Moser bound on the number of maximal independent sets to construct an $(1+\sqrt[3]{3})^n \cdot n^{\mathcal{O}(1)}$ time graph coloring algorithm which was the fastest coloring algorithm for 25 years. In 2003, Eppstein [10] reduced the running time of graph coloring to $\mathcal{O}(2.4151^n)$ by introducing better combinatorial bounds on the number of maximal independent sets of small size. In very recent papers Björklund and Husfeldt [2] reduced the running time to $\mathcal{O}(2.3236^n)$ and finally Björklund-Husfeld and Koivisto to $\mathcal{O}(2^n)$ [3, 23]. Similar example is the work of Byskov and Eppstein [6] who obtain an $\mathcal{O}(2.1020^n)$ time algorithm for deciding if a graph is five colorable which is, again, based on a 1.7724^n combinatorial upper bound on the number of maximal bipartite subgraphs in a graph.

Previous results. Although minimum dominating set is a natural and very interesting problem concerning the design and analysis of exponential-time algorithms, no exact algorithm for it faster than $2^n \cdot n^{O(1)}$ had been known until very recently. In 2004 several different sets of authors obtained algorithms breaking the trivial "2^n-barrier". The algorithm of Fomin, Kratsch, and Woeginger [17] runs in time $\mathcal{O}(1.9379^n)$. The algorithm of Randerath and Schiermeyer [26] uses matching techniques to restrict the search space and runs in time $\mathcal{O}(1.8999^n)$. Grandoni [19, 18], described a $\mathcal{O}(1.8019^n)$ algorithm and finally, Fomin, Grandoni, and Kratsch [12] reduced the running time to $\mathcal{O}(1.5137^n)$. Recently, the running time of this algorithm was improved by van Rooij and Bodlaender to $\mathcal{O}(1.5063^n)$ [29]. All the mentioned results cannot be used to list all the minimal dominating sets. There was no known algorithm listing all minimal dominating sets faster than the trivial time $2^n \cdot n^{\mathcal{O}(1)}$, which is obtained by trying all possible vertex subsets of a graph and check if they are minimal dominating sets. The number of minimal dominating sets is an obvious lower bound on the running time of a listing algorithm. However, besides the trivial $\mathcal{O}(2^n/\sqrt{n})$ bound nothing was known on this number.

There are not so many known exact algorithms for the domatic number. Applying an algorithm similar to Lawler's dynamic programming algorithm [24] to the domatic number problem one obtains an $3^n \cdot n^{\mathcal{O}(1)}$ algorithm. Nothing better was known for this problem. For the three domatic number problem, which is a special case of the domatic number problem where the number of dominating sets is bounded to be at least 3, Reige and Rothe succeed to break the 3^n barrier with an $\mathcal{O}(2.9416^n)$ algorithm [27]. The preliminary version of this paper [15] has triggered a number of results on domatic number. We give a short overview of these results in the concluding section.

Our results. In this paper we construct an algorithm listing all minimal dominating sets in time $\mathcal{O}(1.7159^n)$. The estimation of the running time of the algorithms is based on the inductive proof that the number of minimal dominating sets in a graph on n vertices is at most λ^n, with $\lambda < 1.7159$. The main idea of the proof is inspired by the *Measure and Conquer* technique [12, 13, 14] from exact algorithms, which works as follows. The running time of exponential recursive algorithms is usually bounded by measuring the progress made by the algorithm at each branching step. Though these algorithms may be rather complicated, the measures used in their analysis are often trivial. For example in graph problems the progress is usually measured in terms of number of nodes removed. The idea behind Measure and Conquer is to choose the measure more carefully: a good choice can lead to a tremendous improvement of the running time bounds (for a fixed algorithm). One of the main contributions of this paper is showing that the same basic idea can be successfully applied to derive stronger combinatorial bounds. In particular, the inductive proof of Theorem 4.1 is based on the way we choose the measure of the problem.

In the preliminary version of this paper [15], we obtain an upper bound 1.7697^n on the number of minimal dominating sets. While the main line of the proof is basically the same, the improvement is obtained by a more careful choice of the measure.

Based on the listing algorithm, we derive an $\mathcal{O}(2.8718^n)$ algorithm for the domatic number. The previous fastest domatic number algorithm was an (almost) trivial algorithm of running time $3^n \cdot n^{\mathcal{O}(1)}$.

The rest of this paper is organized as follows. In Section 2 we provide definitions and auxiliary results. In Section 3 the main algorithm listing minimal set covers is given and Section 4 is devoted to the analysis of the algorithm. In Section 5 we describe an algorithm for domatic number and conclude with remarks and open problems in Section 6.

2 Definitions and preliminaries

Let $G = (V, E)$ be a graph. A set $D \subseteq V$ is called a *dominating set* for G if every vertex of G is either in D, or adjacent to some node in D. A dominating set is *minimal* if all its proper subsets are not dominating. We define $\mathbf{DOM}(G)$ to be the number of minimal dominating sets of G.

The *domatic number* $\mathbf{DN}(G)$ of a graph G is the maximum k such that the vertex set $V(G)$ can be split into k pairwise nonintersecting dominating sets. Since every dominating set contains a minimal dominating set, the domatic number $\mathbf{DN}(G)$ can be also defined as the maximum number of pairwise nonintersecting minimal dominating sets in G.

A *Set Cover* instance (SC) is defined by a universe $\mathcal{U}$ of elements and by a collection $\mathcal{S}$ of subsets of $\mathcal{U}$. A *set cover* of $(\mathcal{U}, \mathcal{S})$ is a sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ which covers $\mathcal{U}$, i. e. such that $\cup_{R \in \mathcal{S}'} R = \mathcal{U}$. A set cover is *minimal* if it contains no smaller cover. We denote by $\mathbf{COV}(\mathcal{U}, \mathcal{S})$ the number of minimal set covers of $(\mathcal{U}, \mathcal{S})$.

The problem of listing all minimal dominating sets in G can be naturally reduced to listing all minimal set covers of $(\mathcal{U}, \mathcal{S})$ where $\mathcal{U} = V$ and $\mathcal{S} = \{N[v] \mid v \in V\}$. Note that $N[v] = \{v\} \cup \{u \mid uv \in E\}$ is the set of nodes dominated by v. Thus D is a dominating set of G if and only if $\{N[v] | \ v \in D\}$ is a set cover of $(\mathcal{U}, \mathcal{S})$. In particular, each minimal set cover of $(\mathcal{U}, \mathcal{S})$ corresponds to a minimal dominating set of G.

The following property of minimal set covers is easy to verify.

Lemma 2.1. *Let $\mathcal{S}^*$ be a minimal cover of $(\mathcal{U}, \mathcal{S})$. Then for every subset $R \in \mathcal{S}^*$ at least one of the elements $u \in R$ is covered only by R.*

In the next two sections we show how to compute $\mathbf{DOM}(G)$ in $\mathcal{O}(1.7159^n)$ time. Based on this result, in Section 5 we show how to compute the domatic number of a graph in time $\mathcal{O}(2.8718^n)$.

3 Listing algorithm

In this section we describe a recursive algorithm which lists all the minimal dominating sets of a given input graph $G = (V, E)$. The basic idea is to consider the natural set cover instance $(\mathcal{U}, \mathcal{S}) = (V, \{N[v] \,|\, v \in V\})$ induced by G, and list all the corresponding minimal set covers in it. The set of all minimal dominating sets in G is then easily derived from the set of all minimal set covers of $(\mathcal{U}, \mathcal{S})$.

Since $\mathcal{U}$ changes during the recursion, it is convenient to allow some sets $R \in \mathcal{S}$ to contain elements not in $\mathcal{U}$. Then by *cardinality* $|R|$ of R we mean the number of elements of $\mathcal{U}$ in R, i. e. $|R| = |R \cap \mathcal{U}|$. The *frequency* $|u|$ of $u \in \mathcal{U}$ is the number of subsets $R \in \mathcal{S}$ containing u.

Given an SC instance $(\mathcal{U}, \mathcal{S})$ and a list $\mathcal{C}$ of selected subsets, we next describe a procedure `ListMSC`$(\mathcal{U}, \mathcal{S}, \mathcal{C})$ that lists all the sets of kind $\mathcal{S}^* \cup \mathcal{C}$, where $\mathcal{S}^*$ is a minimal cover of $(\mathcal{U}, \mathcal{S})$. In particular, `ListMSC`$(\mathcal{U}, \mathcal{S}, \emptyset)$ lists all the minimal covers of $(\mathcal{U}, \mathcal{S})$. Procedure `ListMSC`$(\mathcal{U}, \mathcal{S}, \mathcal{C})$ is defined recursively. If $\mathcal{U}$ is empty, the procedure simply returns $\mathcal{C}$. Otherwise, the procedure removes from $\mathcal{S}$ all the sets R of cardinality zero (which cannot belong to any minimal set cover). Now, if $\mathcal{S} = \emptyset$, the procedure halts (there is no set cover at all). In all the other cases the procedure generates one or more subproblems $(\mathcal{U}_i, \mathcal{S}_i, \mathcal{C}_i)$, and solves them recursively.

It remains to describe how the subproblems are generated. There is a sequence of cases. To avoid a confusing nesting of if-then-else statements let us use the following convention: The first case which applies is used first in the procedure. Thus, inside a given case, the hypotheses of all previous cases are assumed to be false. For each case we also provide comments justifying its correctness. When we remove a set R from $\mathcal{S}$, we will say that R is *selected* if we add it to $\mathcal{C}$, and *discarded* otherwise.

Case 0.a: *There is $u \in \mathcal{U}$ of frequency 1.* Let u be covered by $R \in \mathcal{S}$. Every minimal cover must contain R. Hence we branch on the (unique) subproblem obtained by selecting R:

$$(\mathcal{U}_1, \mathcal{S}_1, \mathcal{C}_1) = (\mathcal{U} \setminus R,\, \mathcal{S} \setminus \{R\},\, \mathcal{C} \cup \{R\}). \tag{1}$$

Case 0.b: *There are $u, v \in \mathcal{U}$ such that $\{R \in \mathcal{S} : u \in R\} = \{R \in \mathcal{S} : v \in R\}$.* Consider the problem obtained by removing u from $\mathcal{U}$: Every minimal cover for the new problem is a minimal cover for the original problem and viceversa. Hence we branch on this new problem:

$$(\mathcal{U}_1, \mathcal{S}_1, \mathcal{C}_1) = (\mathcal{U} \setminus \{u\},\, \mathcal{S},\, \mathcal{C}). \tag{2}$$

Case 1: *There is $u \in \mathcal{U}$ belonging only to subsets of cardinality 1.* Let $\mathcal{S}' = \{R_1, R_2, \ldots, R_{|u|}\}$ be the set of all subsets containing u. Note that $|u| \geq 2$ by Case 0. By Lemma 2.1, every minimal cover must contain exactly one of the R_i's. Thus we branch on the $|u|$ subproblems:

$$(\mathcal{U}_i, \mathcal{S}_i, \mathcal{C}_i) = (\mathcal{U} \setminus \{u\},\, \mathcal{S} \setminus \mathcal{S}',\, \mathcal{C} \cup \{R_i\}), \quad i \in \{1, 2, \ldots, |u|\}. \tag{3}$$

Let us denote by $\mathcal{S}_{max}$ the set of all the subsets in $\mathcal{S}$ of maximum cardinality.

Case 2: *There is $R \in \mathcal{S}_{max}$ containing an element of frequency 2.* Let $\mathcal{S}'$ be the set of subsets sharing an element of frequency 2 with R (R excluded). If we discard R, we must select all the sets in $\mathcal{S}'$. Thus we branch on the two subproblems:

$$(\mathcal{U}_1, \mathcal{S}_1, \mathcal{C}_1) = (\mathcal{U} \setminus (\cup_{R' \in \mathcal{S}'} R'),\, \mathcal{S} \setminus (\mathcal{S}' \cup \{R\}),\, \mathcal{C} \cup \mathcal{S}'), \tag{4}$$

$$(\mathcal{U}_2, \mathcal{S}_2, \mathcal{C}_2) = (\mathcal{U} \setminus R,\, \mathcal{S} \setminus \{R\},\, \mathcal{C} \cup \{R\}). \tag{5}$$

Case 3: *There is a set $R \in \mathcal{S}_{max}$, with $|R| \geq 3$.* In this case we simply branch by either selecting or discarding R:

$$(\mathcal{U}_1, \mathcal{S}_1, \mathcal{C}_1) = (\mathcal{U} \setminus R,\, \mathcal{S} \setminus \{R\}, \mathcal{C} \cup \{R\}), \tag{6}$$

$$(\mathcal{U}_2, \mathcal{S}_2, \mathcal{C}_2) = (\mathcal{U},\, \mathcal{S} \setminus \{R\},\, \mathcal{C}). \tag{7}$$

Case 4: *There is $R \in \mathcal{S}_{max}$ which contains another subset $R' \in \mathcal{S}$ (i. e. $R' \subseteq R$).* Also in this case we branch by either selecting or discarding R. Note that when we select R, we can also discard R' (whose cardinality becomes zero). Then:

$$(\mathcal{U}_1, \mathcal{S}_1, \mathcal{C}_1) = (\mathcal{U} \setminus R,\, \mathcal{S} \setminus \{R, R'\},\, \mathcal{C} \cup \{R\}), \tag{8}$$

$$(\mathcal{U}_2, \mathcal{S}_2, \mathcal{C}_2) = (\mathcal{U},\, \mathcal{S} \setminus \{R\},\, \mathcal{C}). \tag{9}$$

Now we can assume that all the hypotheses of Cases 0–4 are false. Observe that then by case 3 the maximum cardinality of a set is 2, and by cases 1 and 4 there are no sets of cardinality 1. Hence all the sets have cardinality exactly 2 (in particular, $\mathcal{S} = \mathcal{S}_{max}$). Therefore, by cases 0 and 2 all the elements have frequency at least 3. Moreover, by case 4 the sets are all distinct.

Case 5: *(All the hypotheses of Cases 0–4 are false).* Let $u \in \mathcal{U}$ be an element of maximum frequency, and $R = \{u, v\} \in \mathcal{S}$ be a subset containing u. Denote by $\mathcal{S}_u = \{R' \in \mathcal{S} \setminus \{R\} \,|\, u \in R'\}$ and $\mathcal{S}_v = \{R' \in \mathcal{S} \setminus \{R\} \,|\, v \in R'\}$ the sets of all the other subsets of $\mathcal{S}$ containing u and v, respectively. By Lemma 2.1, if $\mathcal{S}^*$ is a minimal cover containing R, then either $\mathcal{S}^* \cap \mathcal{S}_u = \emptyset$, or $\mathcal{S}^* \cap \mathcal{S}_v = \emptyset$. This is because otherwise each element of R would be covered by some other set in $\mathcal{S}^*$. In this case we branch on the three subproblems:

$$(\mathcal{U}_1, \mathcal{S}_1, \mathcal{C}_1) = (\mathcal{U} \setminus R,\, \mathcal{S} \setminus (\mathcal{S}_u \cup \{R\}),\, \mathcal{C} \cup \{R\}), \tag{10}$$

$$(\mathcal{U}_2, \mathcal{S}_2, \mathcal{C}_2) = (\mathcal{U} \setminus R,\, \mathcal{S} \setminus (\mathcal{S}_v \cup \{R\}),\, \mathcal{C} \cup \{R\}), \tag{11}$$

$$(\mathcal{U}_3, \mathcal{S}_3, \mathcal{C}_3) = (\mathcal{U},\, \mathcal{S} \setminus \{R\}, \mathcal{C}). \tag{12}$$

4 Bounding the Number of Minimal Dominating Sets

In this section we show that the algorithm of previous section lists all the minimal dominating sets of an n-nodes graph G in time $\mathcal{O}(1.7159^n)$. This implies that the same upper bound holds for the number of minimal dominating sets of G.

As usual, we bound the progress made by the algorithm at each branching step by measuring how much the *size* of the subproblems decreases with respect to the original problem. This leads to a list of recurrences in the size, which can be solved in the standard way. It remains to decide what the *size* of a (sub)problem is. At each branching step we remove some sets and/or elements. So a natural choice for the size of a problem is the total number of sets and elements. (The number of sets would be even more natural, but it leads to a time bound of the kind $\Omega(2^n)$).

Here we measure the size in a refined way. Intuitively, elements of small frequency allow one to use stronger *domination rules*. For example, if there is an element of frequency one, we can remove one set and at least one element while generating a unique subproblem (Case 0). Something similar holds for sets of small cardinality. On the other hand, when we remove a set of large cardinality, we reduce the frequency of many elements. A dual property holds for elements of large frequency. As a consequence, removing sets of large cardinality and elements of large frequency pays of on long term. This kind of phenomenon is not taken into account with the simple measure above, where all the sets and elements have the same *weight*.

This suggests the idea to assign a different weight to elements/sets of different frequency/cardinality. More precisely, we will use the following measure $k(\mathcal{U}, \mathcal{S})$ of the size of a set cover instance $(\mathcal{U}, \mathcal{S})$:

$$k(\mathcal{U}, \mathcal{S}) = \sum_{S \in \mathcal{S}} \alpha_{|S|} + \sum_{u \in \mathcal{U}} \beta_{|u|},$$

where the weights α_i's and β_j's are real numbers to be fixed later. We impose the following constraints on the weights

$$1 \leq \alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 = \alpha_i,\ \forall i \geq 5 \quad \text{and} \quad 0 = \beta_1 < \beta_2 < \beta_3 < \beta_4 < \beta_5 = 1 = \beta_j,\ \forall j \geq 6.$$

The assumption that the weights are an increasing function of the cardinality and of the frequency helps to simplify the analysis. The constraints $\alpha_4 = \alpha_i,\ \forall i \geq 5$ and $\beta_5 = \beta_j,\ \forall j \geq 6$

are only due to computational reasons: we need to compute the weights numerically, and we are not able to do that for an unbounded number of weights. Anyway, trying with some more *free variables* we experimentally observed that the improvement of the bound is negligible. We set $\beta_1 = 0$ since elements of frequency one can be filtered out in polynomial time. Since the result does not change by scaling all the weights by the same positive factor, we also imposed $\beta_5 = 1$.

The following quantities will turn out to be useful in the analysis. Let $\alpha_0 = \beta_0 = 0$. For $i \geq 1$ we denote by

$$\Delta\alpha_i = \alpha_i - \alpha_{i-1} \quad \text{and} \quad \Delta\beta_i = \beta_i - \beta_{i-1}$$

the decrease of the size of the problem after reducing the cardinality of a subset and the frequency of an element from i to $i-1$, respectively. We also define

$$\Delta\alpha_{\leq i} = \min_{1 \leq j \leq i}\{\Delta\alpha_j\} \quad \text{and} \quad \Delta\beta_{\leq i} = \min_{3 \leq j \leq i}\{\Delta\beta_j\}.$$

Note that $\Delta\alpha_{\leq i}$ is the minimum decrease of the size of the problem after reducing by one the cardinality of one set of cardinality at most i. Similarly, $\Delta\beta_{\leq i}$ is the minimum decrease of the size of the problem after reducing by one the frequency of one element whose frequency is between 3 and i. Observe moreover that $\Delta\alpha_i = \Delta\alpha_{\leq i} = 0$ for $i \geq 5$ and $\Delta\beta_i = \Delta\beta_{\leq i} = 0$ for $i \geq 6$. This will allow us to deal with a finite (though very large) number of recurrences only. For the same reason, we imposed $\alpha_1 \geq 1$.

We will sometimes omit $|\cdot|$ in the subscript for notational convenience. For example, we will use α_R for $\alpha_{|R|}$, and $\Delta\beta_u$ for $\Delta\beta_{|u|}$.

Theorem 4.1. *For any graph G on n vertices, all its minimal dominating sets can be listed in time $\mathcal{O}(1.7159^n)$.*

Proof. Consider the listing algorithm described in previous section. With a slight notational abuse, here we denote by $\mathbf{COV}(\mathcal{U}, \mathcal{S})$ the number of set covers listed by `ListMSC`$(\mathcal{U}, \mathcal{S}, \mathcal{C})$ (the value of $\mathcal{C}$ does not influence such number). Clearly, $\mathbf{COV}(\mathcal{U}, \mathcal{S})$ is an upper bound on the number of minimal set covers of $(\mathcal{U}, \mathcal{S})$ (all the minimal set covers are listed, but some of them may be listed several times). Observe that the running time of the algorithm is within a polynomial factor from $\mathbf{COV}(\mathcal{U}, \mathcal{S})$.

Let $k = k(\mathcal{U}, \mathcal{S})$ be the size of the problem, according to the measure described above. By $\mathbf{COV}(k)$ we denote the maximum value of $\mathbf{COV}(\mathcal{U}, \mathcal{S})$ among all the `SC` instances $(\mathcal{U}, \mathcal{S})$ of size k. We will show that $\mathbf{COV}(k) \leq \lambda^k$, for some constant $\lambda > 1$ which depends on the weights. We proceed by induction on k. Clearly, $\mathbf{COV}(0) \leq 1$. Suppose that $\mathbf{COV}(k') \leq \lambda^{k'}$ for every $0 \leq k' < k$, and consider any instance $(\mathcal{U}, \mathcal{S})$ of size k. Clearly, if the procedure halts on $(\mathcal{U}, \mathcal{S})$, $\mathbf{COV}(\mathcal{U}, \mathcal{S}) \leq 1 \leq \lambda^k$. Thus let us consider all the possible branchings, following the structure of the algorithm.

Case 0.a: *There is $u \in \mathcal{U}$ of frequency 1.* Recall that $R \ni u$. Let $(\mathcal{U}_1, \mathcal{S}_1)$ be the `SC` instance given by (1). Since

$$k(\mathcal{U}_1, \mathcal{S}_1) \leq k - \alpha_R \leq k - \alpha_1,$$

by the inductive hypothesis:

$$\mathbf{COV}(\mathcal{U},\mathcal{S}) \le \mathbf{COV}(k(\mathcal{U}_1,\mathcal{S}_1)) \le \lambda^{k-\alpha_1}. \tag{13}$$

By Case 0.a, from now on we can assume $|u| \ge 2$ for every element u.

Case 0.b: *There are $u, v \in \mathcal{U}$ such that $\{R \in \mathcal{S} : u \in R\} = \{R \in \mathcal{S} : v \in R\}$.* Let $(\mathcal{U}_1, \mathcal{S}_1)$ be the `SC` instance given by (2). Recall that $\mathcal{U}_1 = \mathcal{U} \setminus \{u\}$. By Case 0.a, $|u| \ge 2$. Then

$$k(\mathcal{U}_1,\mathcal{S}_1) \le k - \beta_u \le k - \beta_2,$$

and by the inductive hypothesis:

$$\mathbf{COV}(\mathcal{U},\mathcal{S}) \le \mathbf{COV}(k(\mathcal{U}_1,\mathcal{S}_1)) \le \lambda^{k-\beta_2}. \tag{14}$$

Case 1: *There is $u \in \mathcal{U}$ belonging only to subsets of cardinality 1.* Consider the subproblems $(\mathcal{U}_i, \mathcal{S}_i)$, $i \in \{1, 2, \ldots, |u|\}$, given by (3). In each subproblem we remove the element u together with all the $|u|$ subsets containing it. Observe that $|u| \ge 2$ by Case 0. Thus

$$\mathbf{COV}(\mathcal{U},\mathcal{S}) \le \sum_{i=1}^{|u|} \mathbf{COV}(k(\mathcal{U}_i,\mathcal{S}_i)) \le |u| \cdot \mathbf{COV}(k - |u|\,\alpha_1 - \beta_u) \le |u|\lambda^{k-|u|\alpha_1-\beta_u}.$$

For $|u| \ge 5$, by assumption $\beta_u = 1$. Moreover, by the assumption $\alpha_1 \ge 1$,

$$\max_{|u|\ge 5}\{|u|\lambda^{-|u|\alpha_1-1}\} = 5\lambda^{-5\alpha_1-1}.$$

Hence, in order to show that $\mathbf{COV}(\mathcal{U},\mathcal{S}) \le \lambda^k$, we can restrict our attention to the following set of recurrences

$$\forall\, 2 \le r \le 5 :\ \mathbf{COV}(\mathcal{U},\mathcal{S}) \le r\lambda^{k-r\,\alpha_1-\beta_r}. \tag{15}$$

By Case 1, from now on every $R \in \mathcal{S}_{max}$ satisfies $|R| \ge 2$.

Observe that, by the assumptions on the weights, for a set R and an element u, $\alpha_R := \alpha_{|R|} = \alpha_{\min\{|R|,5\}}$ and $\beta_R := \beta_{|R|} = \beta_{\min\{|u|,6\}}$. By the same argument, $\Delta\alpha_R = \Delta\alpha_{\min\{|R|,5\}}$ and $\Delta\beta_R = \Delta\beta_{\min\{|u|,6\}}$. This observation will be crucial to obtain a finite number of recurrences in the following cases.

Case 2: *There is $R \in \mathcal{S}_{max}$ containing an element of frequency* 2. In this case

$$\mathbf{COV}(\mathcal{U},\mathcal{S}) \le \mathbf{COV}(k(\mathcal{U}_1,\mathcal{S}_1)) + \mathbf{COV}(k(\mathcal{U}_2,\mathcal{S}_2)),$$

where $(\mathcal{U}_1,\mathcal{S}_1)$ and $(\mathcal{U}_2,\mathcal{S}_2)$ are given by (4) and (5). Let us upper bound the size of problems $(\mathcal{U}_1,\mathcal{S}_1)$ and $(\mathcal{U}_2,\mathcal{S}_2)$. Recall that $\mathcal{S}'$ is the set of subsets sharing an element of frequency 2 with R. In the subproblem $(\mathcal{U}_1,\mathcal{S}_1)$ we remove R, all the sets in $\mathcal{S}'$, and all the elements $\mathcal{U}' = \cup_{R' \in \mathcal{S}'} R'$ covered by $\mathcal{S}'$. Moreover we reduce by one the frequency of all the elements of $R \setminus \mathcal{U}'$. In the subproblem $(\mathcal{U}_2,\mathcal{S}_2)$ we remove R and its elements. Moreover, for each $u \in R$, the cardinality of $(|u|-1)$ sets is decreased by one. Since $R \in \mathcal{S}_{max}$, each time we remove one element from a set, the reduction of the size of the problem is at least $\Delta\alpha_{\leq |R|} = \Delta\alpha_{\leq \min\{|R|,5\}}$ (since $|R|$ is the maximum cardinality of any set). More precisely, if the cardinality of a given set is reduced from i to j, then the size of the problem decreases by $\alpha_i - \alpha_j \geq (j-i)\Delta\alpha_{\leq |R|}$. In order to simplify the analysis, we distinguish two subcases.

Case 2.a: *At least one element u of frequency 2 is contained in a subset $R' \in \mathcal{S}$ such that $R' \subseteq R$.* In subproblem $(\mathcal{U}_1, \mathcal{S}_1)$ we remove both subsets R and R', and all the elements of R'. Let $r = \min\{|R|, 5\} \in [2, 5]$ and $r' = \min\{|R'|, 5\} \in [1, r]$. Observe that, for any element u, $\beta_u \geq \beta_2$ (since the minimum frequency is 2 and β_i is non-decreasing in i). Thus

$$k(\mathcal{U}_1, \mathcal{S}_1) \leq k - \alpha_R - \alpha_{R'} - \sum_{u \in R'} \beta_u \leq k - \alpha_R - \alpha_{R'} - |R'|\beta_2 \leq k - \alpha_r - \alpha_{r'} - r'\beta_2.$$

In subproblem $(\mathcal{U}_2, \mathcal{S}_2)$ we remove both subsets R and R', and all the elements of R. By the same arguments as above we get:

$$k(\mathcal{U}_2, \mathcal{S}_2) \leq k - \alpha_R - \alpha_{R'} - \sum_{u \in R} \beta_u \leq k - \alpha_r - \alpha_{r'} - r\beta_2.$$

Altogether, we obtain the following set of recurrences

$$\forall\, 2 \leq r \leq 5,\ 1 \leq r' \leq r: \quad \mathbf{COV}(\mathcal{U}, \mathcal{S}) \quad \leq \quad \lambda^{k-\alpha_r-\alpha_{r'}-r'\beta_2} + \lambda^{k-\alpha_r-\alpha_{r'}-r\beta_2}. \qquad (16)$$

Case 2.b: *There is no $R' \in \mathcal{S}$ containing an element of frequency 2 such that $R' \subseteq R$.* Let $R|_r$ be the $r = \min\{|R|, 5\}$ elements of R of smallest frequency. For $2 \leq i \leq 5$, we denote by r_i the number of elements of $R|_r$ of frequency i, and by r_6 the number of the remaining elements of $R|_r$ (of frequency 6 *or larger*). Note that $\sum_{i=1}^{6} r_i = \sum_{i=2}^{6} r_i = r \leq 5$. Observe also that each element of R of frequency 2 lies in a distinct set from $\mathcal{S}'$. In fact, otherwise there would be two elements u and v, $|u| = |v| = 2$, belonging to R and to some $R' \in \mathcal{S}'$; this is excluded by Case 0.b. Therefore $|\mathcal{S}'| \geq r_2$. We eventually observe that there is at least one element $v \in \mathcal{U}' \setminus R$. Hence

$$\begin{aligned}
k(\mathcal{U}_1, \mathcal{S}_1) &\leq k - \alpha_R - \sum_{R' \in \mathcal{S}'} \alpha_{R'} - \sum_{u \in \mathcal{U}'} \beta_u - \sum_{u \in R \setminus \mathcal{U}'} \Delta\beta_u \\
&\leq k - \alpha_R - |\mathcal{S}'|\alpha_2 - \beta_v - \sum_{u \in R \cap \mathcal{U}'} \beta_u - \sum_{u \in R \setminus \mathcal{U}'} \Delta\beta_u \\
&\leq k - \alpha_R - r_2\alpha_2 - \beta_v - \sum_{u \in R \cap \mathcal{U}'} \Delta\beta_u - \sum_{u \in R \setminus \mathcal{U}'} \Delta\beta_u \\
&= k - \alpha_R - r_2\alpha_2 - \beta_v - \sum_{u \in R} \Delta\beta_u \\
&\leq k - \alpha_r - r_2\alpha_2 - \beta_2 - \sum_{i=2}^{5} r_i \Delta\beta_i,
\end{aligned}$$

where we used the fact that $\beta_u \geq \Delta\beta_u$ and $\Delta\beta_u = 0$ for $|u| \geq 6$. Analogously, observing that $\Delta\alpha_{\leq j} = \Delta\alpha_j = 0$ for $j \geq 5$ and hence $\Delta\alpha_{\leq |R|} = \Delta\alpha_{\leq r}$, and recalling that $\beta_5 = \beta_i$ for $i \geq 6$,

$$k(\mathcal{U}_2, \mathcal{S}_2) \leq k - \alpha_R - \sum_{u \in R} \beta_u - \sum_{u \in R} (|u| - 1)\Delta\alpha_{\leq |R|}$$
$$\leq k - \alpha_r - \sum_{i=2}^{6} r_i \beta_i - \sum_{i=2}^{6} r_i (i-1)\Delta\alpha_{\leq r}.$$

Altogether,

$$\forall\, 2 \leq r \leq 5,\ r_i \geq 0,\ r_2 \geq 1,\ \sum_{i=2}^{6} r_i = r:\ \mathbf{COV}(\mathcal{U}, \mathcal{S}) \leq \lambda^{k - \alpha_r - r_2\alpha_2 - \beta_2 - \sum_{i=2}^{5} r_i \Delta\beta_i}$$
$$+ \lambda^{k - \alpha_r - \sum_{i=2}^{6} r_i\beta_i - \sum_{i=2}^{6} r_i(i-1)\Delta\alpha_{\leq r}}. \quad (17)$$

By Case 2, in the following we can assume $|u| \geq 3$ for every $u \in R \in \mathcal{S}_{max}$.

Case 3: *There is a set $R \in \mathcal{S}_{max}$, with $|R| \geq 3$.* Let $(\mathcal{U}_1, \mathcal{S}_1)$ and $(\mathcal{U}_2, \mathcal{S}_2)$ be given by (6) and (7). In the subproblem $(\mathcal{U}_1, \mathcal{S}_1)$ we remove R and we decrease the frequency of its elements by one. Hence, with the usual notation,

$$k(\mathcal{U}_1, \mathcal{S}_1) \leq k - \alpha_R - \sum_{u \in R} \Delta\beta_u \leq k - \alpha_r - \sum_{i=3}^{5} r_i \Delta\beta_i.$$

In the subproblem $(\mathcal{U}_2, \mathcal{S}_2)$ we remove R and its elements. Moreover, for each $u \in R$, the cardinality of $(|u| - 1)$ sets is decreased by one:

$$k(\mathcal{U}_2, \mathcal{S}_2) \leq k - \alpha_R - \sum_{u \in R} \beta_u - \sum_{u \in R} (|u| - 1)\Delta\alpha_{\leq |R|} \leq k - \alpha_r - \sum_{i=3}^{6} r_i \beta_i - \sum_{i=3}^{6} r_i (i-1)\Delta\alpha_{\leq r}.$$

We can conclude that

$$\forall\, 3 \leq r \leq 5,\ r_i \geq 0,\ \sum_{i=3}^{6} r_i = r:\ \mathbf{COV}(\mathcal{U}, \mathcal{S}) \leq \lambda^{k - \alpha_r - \sum_{i=3}^{5} r_i \Delta\beta_i}$$
$$+ \lambda^{k - \alpha_r - \sum_{i=3}^{6} r_i\beta_i - \sum_{i=3}^{6} r_i(i-1)\Delta\alpha_{\leq r}}. \quad (18)$$

By Cases 1 and 3, we can assume in the following that $|R| = 2$ for all $R \in \mathcal{S}_{max}$.

Case 4: *There is $R \in \mathcal{S}_{max}$ which contains another subset $R' \in \mathcal{S}$ ($R' \subseteq R$).* Let $(\mathcal{U}_1, \mathcal{S}_1)$ and $(\mathcal{U}_2, \mathcal{S}_2)$ be given by (8) and (9). In subproblem $(\mathcal{U}_1, \mathcal{S}_1)$ we remove $R = \{u, v\}$ and we decrease by one the cardinality of its elements. Define $u' = \min\{|u|, 6\} \in [3, 6]$ and $v' = \min\{|v|, 6\} \in [3, 6]$. Note that $\beta_u = \beta_{u'}$, and hence $\Delta\beta_u = \Delta\beta_{u'}$. Analogously, $\beta_v = \beta_{v'}$ and $\Delta\beta_v = \Delta\beta_{v'}$. Therefore,

$$k(\mathcal{U}_1, \mathcal{S}_1) \leq k - \alpha_R - \sum_{u \in R} \Delta\beta_u = k - \alpha_2 - \Delta\beta_{u'} - \Delta\beta_{v'}.$$

In subproblem $(\mathcal{U}_2, \mathcal{S}_2)$ we remove R, R', and the elements of R. Moreover, for each $w \in R$, we decrease by one the cardinality of $(|w|-1)$ sets if $w \in R \setminus R'$ and of $(|w|-2)$ sets if $w \in R'$. Then, for $r' = |R'| \in [1, 2]$,

$$\begin{aligned} k(\mathcal{U}_2, \mathcal{S}_2) &\le k - \alpha_R - \alpha_{R'} - \sum_{w \in R} \beta_w - \sum_{w \in R \setminus R'} (|w|-1)\Delta\alpha_{\le |R|} - \sum_{w \in R'} (|w|-2)\Delta\alpha_{\le |R|} \\ &= k - \alpha_2 - \alpha_{r'} - \beta_{u'} - \beta_{v'} - (|u| + |v| - 2 - r')\Delta\alpha_{\le 2} \\ &\le k - \alpha_2 - \alpha_{r'} - \beta_{u'} - \beta_{v'} - (u' + v' - 2 - r')\Delta\alpha_{\le 2}. \end{aligned}$$

Altogether,

$$\begin{aligned} \forall\, 1 \le r' \le 2,\ 3 \le u' \le 6,\ 3 \le v' \le 6:\ \mathbf{COV}(\mathcal{U}, \mathcal{S}) &\le \lambda^{k - \alpha_2 - \Delta\beta_{u'} - \Delta\beta_{v'}} \\ &+ \lambda^{k - \alpha_2 - \alpha_{r'} - \beta_{u'} - \beta_{v'} - (u' + v' - 2 - r')\Delta\alpha_{\le 2}}. \end{aligned} \tag{19}$$

Case 5: *(All the hypotheses of Cases 0–4 are false).* In this case

$$\mathbf{COV}(\mathcal{U}, \mathcal{S}) \le \mathbf{COV}(\mathcal{U}_1, \mathcal{S}_1) + \mathbf{COV}(\mathcal{U}_2, \mathcal{S}_2) + \mathbf{COV}(\mathcal{U}_3, \mathcal{S}_3),$$

where $(\mathcal{U}_1, \mathcal{S}_1)$, $(\mathcal{U}_2, \mathcal{S}_2)$, and $(\mathcal{U}_3, \mathcal{S}_3)$, are given by (10), (11), and (12). In the first two subproblems we remove $R = \{u, v\}$, its elements, and we either remove $\mathcal{S}_u$ or $\mathcal{S}_v$. Recall that $\mathcal{S}_u \subseteq \mathcal{S}_{max}$ ($\mathcal{S}_v \subseteq \mathcal{S}_{max}$) is the collection of subsets whose intersection with R is $\{u\}$ ($\{v\}$). By Case 4, R cannot contain any other set. As a consequence, $|\mathcal{S}_u| = |u| - 1$ and $|\mathcal{S}_v| = |v| - 1$. By the same argument, we also know that any two sets of $\mathcal{S}_u$ ($\mathcal{S}_v$) do not share any element besides $\{u\}$ ($\{v\}$): in fact, otherwise the two sets would be contained in each other. It follows that, when we remove $\mathcal{S}_u$ ($\mathcal{S}_v$), we decrease exactly by one the frequency of each element $w \neq u$ in $\mathcal{S}_u$ ($w \neq u$ in $\mathcal{S}_v$). Since $\mathcal{S}_u$ and $\mathcal{S}_v$ belong to $\mathcal{S}_{max}$, and any element in $\mathcal{S}_{max}$ has frequency between 3 and $|u|$ by the previous cases, the consequent reduction of the size of the problem is $\Delta\beta_w \ge \Delta\beta_{\le |u|} := \min_{3 \le j \le |u|}\{\Delta\beta_j\}$. Defining $u' = \min\{|u|, 6\} \in [3, 6]$ and $v' = \min\{|v|, 6\} \in [3, u']$, we obtain

$$\begin{aligned} k(\mathcal{U}_1, \mathcal{S}_1) &\le k - \alpha_R - \sum_{R' \in \mathcal{S}_u} \alpha_{R'} - \sum_{w \in R} \beta_w - \sum_{R' \in \mathcal{S}_v} \Delta\alpha_{R'} - \sum_{w \in R' \setminus R, R' \in \mathcal{S}_u} \Delta\beta_w \\ &\le k - \alpha_2 - (|u| - 1)\alpha_2 - \beta_u - \beta_v - (|v| - 1)\Delta\alpha_2 - (|u| - 1)\Delta\beta_{\le |u|} \\ &\le k - u'\,\alpha_2 - \beta_{u'} - \beta_{v'} - (v' - 1)\Delta\alpha_2 - (u' - 1)\Delta\beta_{\le u'}. \end{aligned}$$

Similarly,

$$k(\mathcal{U}_2, \mathcal{S}_2) \le k - v'\,\alpha_2 - \beta_{u'} - \beta_{v'} - (u' - 1)\Delta\alpha_2 - (v' - 1)\Delta\beta_{\le u'}.$$

In the last subproblem we remove R, and we decrease by one the frequencies of its elements:

$$k(\mathcal{U}_3, \mathcal{S}_3) \le k - \alpha_R - \sum_{w \in R} \Delta\beta_w = k - \alpha_2 - \Delta\beta_u - \Delta\beta_v = k - \alpha_2 - \Delta\beta_{u'} - \Delta\beta_{v'}.$$

Altogether, for all u', v' such that $3 \le u' \le 6$ and $3 \le v' \le u'$, we have that

$$\begin{aligned}\mathbf{COV}(\mathcal{U}, \mathcal{S}) \le\; & \lambda^{k-u'\,\alpha_2-\beta_{u'}-\beta_{v'}-(v'-1)\Delta\alpha_2-(u'-1)\Delta\beta_{\le u'}} \\ & + \lambda^{k-v'\,\alpha_2-\beta_{u'}-\beta_{v'}-(u'-1)\Delta\alpha_2-(v'-1)\Delta\beta_{\le u'}} \\ & + \lambda^{k-\alpha_2-\Delta\beta_{u'}-\Delta\beta_{v'}}. \qquad (20)\end{aligned}$$

Summarizing from Cases 0–5 we obtain Recurrences (13)–(20). For given values of α's and β's, we wish to find a value of λ (the smallest possible) such that the righthand sides of the inequalities (13)–(20) are upper bounded by λ^k. The value of λ is a function of the weights α's and β's, and computing the best weights is an interesting optimization problem in its own. We refer to Eppstein's work [11] on quasi-convex programming which provides a general framework for solving this type of problems.

We obtained numerically the following feasible values of the weights: $\alpha_1 = 2.328463$, $\alpha_2 = 2.505092$, $\alpha_3 = 2.670885$, $\alpha_4 = 2.720886$, $\beta_2 = 0.120153$, $\beta_3 = 0.772504$, $\beta_4 = 0.973506$. For these values of the weights, a feasible value of λ is $\lambda = 1.156154$. Recall that, in order to check that $\lambda = 1.156154$ is feasible, it is sufficient to check that the righthand sides of Inequalities (13)–(20) are upper bounded by λ^k. For example, let us consider Recurrence 18 with $r = 4$, $r_3 = 2$, $r_4 = 1$ and $r_6 = 1$. We have $\Delta\beta_3 = 0.652351$, $\Delta\beta_4 = 0.201002$, and $\Delta\alpha_{\le 4} = \min\{\Delta\alpha_1, \Delta\alpha_2, \Delta\alpha_3, \Delta\alpha_4\} = 0.050001$. Thus, we need to check that

$$\begin{aligned}\lambda^k &\ge \lambda^{k-\alpha_r-\sum_{i=3}^{5} r_i\Delta\beta_i} + \lambda^{k-\alpha_r-\sum_{i=3}^{6} r_i\beta_i-\sum_{i=3}^{6} r_i(i-1)\Delta\alpha_{\le r}} \\ &= \lambda^{k-\alpha_4-2\Delta\beta_3-\Delta\beta_4} + \lambda^{k-\alpha_4-2\beta_3-\beta_4-\beta_6-(2\cdot 1+1\cdot 3+1\cdot 5)\Delta\alpha_{\le 4}} \\ &= \lambda^{k-2.720886-2\cdot 0.652351-0.201002} + \lambda^{k-2.720886-2\cdot 0.772504-0.973506-1-10\cdot 0.050001} \\ &= \lambda^{k-4.22659} + \lambda^{k-6.73941}.\end{aligned}$$

Since

$$\lambda^{-4.22659} + \lambda^{-6.73941} = 1.156154^{-4.22659} + 1.156154^{-6.73941} < 0.9177 < 1,$$

we have that

$$\lambda^k \ge \lambda^{k-4.22659} + \lambda^{k-6.73941}.$$

While checking by hand all inequalities is tedious, by making use of a simple program the reader may verify that also the other inequalities are satisfied (the pseudo-code of this program is provided in the Appendix).

For any graph G on n vertices the size of the corresponding `SC` instance $(\mathcal{U}, \mathcal{S})$ is at most $\beta_5|\mathcal{U}| + \alpha_4|\mathcal{S}| = (1 + \alpha_4)\,n$. It follows that

$$\mathbf{DOM}(G) \le \mathbf{COV}((1 + \alpha_4)n) < 1.156154^{(1+2.720886)n} < 1.7159^n.$$

This completes the proof. □

We remark that, in the proof above, any feasible choice of the weights and of λ provides an upper bound on $\mathbf{DOM}(G)$ (though possibly not the best possible). Hence, the mentioned values and the set of recurrences is all one needs to check the correctness of the claim. We also remark that using a real-valued induction variable is crucial with our approach: Restricting ourselves to integral variables would either weaken the bounds obtained or make the analysis considerably more complicated. Note that the correctness of the induction is guaranteed by the fact that the sizes are always non-negative by definition and that the size of each subproblem decreases at least by a constant positive quantity (with respect to the size of the original problem).

Theorem 4.1 implies the following combinatorial result which is interesting on its own.

Corollary 4.2. *Every graph on n vertices contains at most 1.7159^n minimal dominating sets.*

5 Computing the domatic number

The algorithm listing minimal set covers (and minimal dominating sets) can be used to compute the domatic number of a graph $G = (V, E)$. Our algorithm has similarities with the classical algorithm computing the chromatic number due to Lawler [24] (see also [10]), but the analysis of our algorithm is based on Theorem 4.1.

For every set $X \subseteq V$ denote by $\mathbf{DN}(G|X)$ the maximum number of pairwise nonintersecting subsets of X such that each of these subsets is a minimal dominating set in G. Clearly, $\mathbf{DN}(G|V) = \mathbf{DN}(G)$ is the domatic number of G. Note that if X is not dominating, then $\mathbf{DN}(G|X) = 0$

We use an array A, indexed by the 2^n subsets X of V, for which we compute the numbers $\mathbf{DN}(G|X)$. Initially the array is set to zero. Then we run through the subsets X of V, in increasing cardinality order. To compute $A[X]$, we run through all minimal dominating sets $D \subseteq X$ of G, and put

$$A[X] = \max\{A[X \setminus D] + 1 \mid D \subseteq X \text{ and } D \text{ is a minimal dominating set in } G\}.$$

Finally, after running through all subsets, we return the value in $A[V]$ as the domatic number of G.

Theorem 5.1. *The domatic number of a graph G on n vertices can be computed in time $O(2.8718^n)$.*

Proof. The correctness of the algorithm `DN` can be shown by an easy induction. Let X be a subset of V. Suppose that after running the algorithm, for every proper subset S of X the value $A[S]$ is equal to $\mathbf{DN}(G|S)$. Note that $A[\emptyset] = 0$. If X contains no dominating subsets (i. e. X is not dominating), then we have that $A[X] = 0 = \mathbf{DN}(G|X)$. Otherwise, $\mathbf{DN}(G|X)$ is equal to $\max\{\mathbf{DN}(G|(X \setminus D)) + 1\}$, where the maximum is taken over all the minimal dominating sets $D \subseteq X$, and thus the value $A[X]$ computed by the algorithm is equal to $\mathbf{DN}(G|X)$.

For a set $X \subseteq V$, let $\mathbf{DOM}(G|X)$ be the number of minimal dominating sets of G which are subsets of X. To estimate the running time of the algorithm, let us bound first $\mathbf{DOM}(G|X)$. We consider the following modified reduction to `SC`: $\mathcal{U} = V$ and $\mathcal{S} = \{N[v] \mid v \in X\}$. Then $\mathbf{DOM}(G|X) = \mathbf{COV}(\mathcal{U}, \mathcal{S})$. Note that the size of this problem is at most $\beta_5|\mathcal{U}| + \alpha_4|\mathcal{S}| = n + \alpha_4|X|$. By the proof of Theorem 4.1, $\mathbf{COV}(\mathcal{U}, \mathcal{S}) \leq \mathbf{COV}(n + \alpha_4\,|X|) \leq$

$\lambda^{n+\alpha_4|X|}$, where the values of the α_i's, β_j's, and of λ must satisfy the constraint that the righthand sides of Recurrences (13)–(20) is upper bounded by λ^k. Using the listing algorithm of Section 3, one can list in time $\mathcal{O}(\lambda^{n+\alpha_4|X|})$ all the minimal dominating sets contained in X. Hence the running time of the algorithm is bounded by

$$\mathcal{O}(n^{O(1)}\sum_{i=0}^{n}\binom{n}{i}\lambda^{n+\alpha_4 i}) = \mathcal{O}(\lambda^n(1+\lambda^{\alpha_4})^n n^{O(1)}).$$

We numerically found the following values for the weights: $\alpha_1 = 2.450844$, $\alpha_2 = 2.692202$, $\alpha_3 = 2.856464$, $\alpha_4 = 2.924811$, $\beta_2 = 0.120446$, $\beta_3 = 0.756800$, $\beta_4 = 0.960327$, and $\lambda = 1.148698$. Then $\mathcal{O}(\lambda^n(1+\lambda^{\alpha_4})^n\, n^{O(1)}) = \mathcal{O}(2.8718^n)$. This concludes the proof. □

6 Conclusions and open problems

Using Measure and Conquer, we have shown that minimal dominating sets of a graph on n vertices can be listed in time $O(1.7159^n)$. We think that this work opens an interesting line of research, namely using Measure and Conquer to list combinatorial objects and to develop new combinatorial lower and upper bounds.

As an application of our listing algorithm we obtained a faster exponential algorithm to compute the domatic number of a graph. Our algorithm runs in time $\mathcal{O}(2.8718^n)$, thus improving on the trivial $3^n \cdot n^{\mathcal{O}(1)}$ algorithm, and even on the $\mathcal{O}(2.9416^n)$ algorithm in [27] for the special case of three domatic number. The preliminary version of this paper [15], where we obtained the upper bound 1.7697^n on the number of minimal dominating sets, has been used in some other works on domatic number. Chiniforooshan and Simjour [7] modified the arguments from [15] for a number of minimal dominating sets of small size and thus obtain an algorithm computing three domatic number in time $\mathcal{O}(2.7393^n)$. Riege et al. [28] obtained an $\mathcal{O}(2.695^n)$ time algorithm for three domatic number by combining the result from [15] with techniques for SAT. Björklund, Husfeldt, and Koivisto [3, 23] used [15] to obtain polynomial space algorithm with running time $\mathcal{O}(2.8805^n)$. Björklund, Husfeldt, and Koivisto [3, 23] also observed how to compute the domatic number of a graph in time $2^n \cdot n^{\mathcal{O}(1)}$ and exponential space. Let us note, that while our Lawler type algorithm for domatic number uses exponential space, by making use of the technique of Björklund, Husfeldt and Koivisto [4] it can be turned into a polynomial space algorithm.

Let us conclude with some open questions.

- **Lower and upper bounds.** Dieter Kratsch (private communication) found graphs ($n/6$ disjoint copies of the octahedron) containing $15^{n/6} \approx 1.5704^n$ minimal dominating sets. We conjecture that Kratsch's graphs are the graphs with the maximum number of minimal dominating sets. This suggests the possibility that our bound for the maximum number of minimal dominating sets might be further improved. Finding a tighter bound, possibly by means of a refined measure, is an interesting open problem, both from a combinatorial and from an algorithmic point of view.

- **Listing with polynomial time delay.** There is a number of listing algorithms in the literature that list combinatorial objects with a polynomial time delay. The existence of such an algorithm for minimal dominating sets is an interesting open problem.

Acknowledgement. We are grateful to Dieter Kratsch for bringing our attention to the problem and for many fruitful discussions.

References

[1] R. Beigel, *Finding maximum independent sets in sparse and general graphs*, in Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms (SODA 1999), ACM and SIAM, 1999, pp. 856–857.

[2] A. Björklund and T. Husfeldt, *Exact algorithms for exact satisfiability and number of perfect matchings*, in Proceedings of the 33nd International Colloquium on Automata, Languages and Programming (ICALP 2006), Lecture Notes in Comput. Sci., Springer, Berlin, 2006.

[3] ——, *Inclusion–Exclusion Algorithms for Counting Set Partitions*, in the proceedings of FOCS'06, 575–582 (2006).

[4] A. Björklund, T. Husfeldt, and M. Koivisto, *Set partitioning via inclusionexclusion*, SIAM J. Computing, to appear.

[5] L. M. Brègman, *Certain properties of nonnegative matrices and their permanents*, Doklady Akademii Nauk BSSR, 211 (1973), pp. 27–30.

[6] J. M. Byskov and D. Eppstein, *An algorithm for enumerating maximal bipartite subgraphs*, manuscript, (2004).

[7] E. Chiniforooshan and N. Simjour, *An* $O(2.7393^n)$ *algorithm for the three domatic number problem*, manuscript, 2006.

[8] E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, P. Raghavan, and U. Schöning, *A deterministic* $(2 - 2/(k+1))^n$ *algorithm for k-SAT based on local search*, Theoretical Computer Science, 289 (2002), pp. 69–83.

[9] G. P. Egorychev, *Proof of the van der Waerden conjecture for permanents*, Sibirsk. Mat. Zh., 22 (1981), pp. 65–71, 225.

[10] D. Eppstein, *Small maximal independent sets and faster exact graph coloring*, Journal of Graph Algorithms and Applications, 7 (2003), pp. 131–140.

[11] D. Eppstein, *Quasiconvex analysis of multivariate recurrence equations for backtracking algorithms*, ACM Transactions on Algorithms, 2006 (2), pp. 492-509.

[12] F. V. Fomin, F. Grandoni, and D. Kratsch, *Measure and conquer: Domination – a case study*, in Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005), vol. 3580 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 191–203.

[13] ——, *Some new techniques in design and analysis of exact (exponential) algorithms*, Bulletin of the EATCS, 87 (2005), pp. 47–77.

[14] F. V. Fomin, F. Grandoni, and D. Kratsch, *Measure and conquer: A simple $O(2^{0.288\,n})$ independent set algorithm*, in 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006), New York, 2006, ACM and SIAM, pp. 18–25.

[15] F. V. Fomin, F. Grandoni, A. V. Pyatkin, and A. Stepanov, *Bounding the number of minimal dominating sets: a measure and conquer approach*, in Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005), vol. 3827 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 573–582.

[16] F. V. Fomin, D. Kratsch, and I. Todinca, *Exact algorithms for treewidth and minimum fill-in*, in Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004), vol. 3142 of Lecture Notes in Comput. Sci., Springer, Berlin, 2004, pp. 568–580.

[17] F. V. Fomin, D. Kratsch, and G. J. Woeginger, *Exact (exponential) algorithms for the dominating set problem*, in Proceedings of the 30th Workshop on Graph Theoretic Concepts in Computer Science (WG 2004), vol. 3353 of Lecture Notes in Comput. Sci., Springer, Berlin, 2004, pp. 245–256.

[18] F. Grandoni, *A note on the complexity of minimum dominating set*, Journal of Discrete Algorithms, 4 (2006), pp. 209–214.

[19] ——, *Exact algorithms for hard graph problems*, PhD thesis, Università di Roma "Tor Vergata", Roma, Italy, (March, 2004).

[20] T. W. Haynes and S. T. Hedetniemi, eds., *Domination in graphs*, Marcel Dekker Inc., New York, 1998.

[21] M. Held and R. M. Karp, *A dynamic programming approach to sequencing problems*, Journal of SIAM, 10 (1962), pp. 196–210.

[22] J. Iwama and S. Tamaki, *Improved upper bounds for 3-sat*, in 15th ACM-SIAM Symposium on Discrete Algorithms (SODA 2004), ACM and SIAM, 2004, p. 328.

[23] M. Koivisto, An $O(2^n)$ Algorithm for Graph Colouring and Other Partitioning Problems via Inclusion-Exclusion, in the proceedings of FOCS'06, 583–590 (2006).

[24] E. L. Lawler, *A note on the complexity of the chromatic number problem*, Information Processing Lett., 5 (1976), pp. 66–67.

[25] J. W. Moon and L. Moser, *On cliques in graphs*, Israel Journal of Mathematics, 3 (1965), pp. 23–28.

[26] B. Randerath and I. Schiermeyer, *Exact algorithms for MINIMUM DOMINATING SET*, Technical Report zaik-469, Zentrum für Angewandte Informatik Köln, Germany, 2004.

[27] T. Reige and J. Rothe, *An exact* 2.9416^n *algorithm for the three domatic number problem*, in Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS 2005), vol. 3618 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 733–744.

[28] T. Riege, J. Rothe, H. Spakowski, and M. Yamamoto, *An improved exact algorithm for the domatic number problem*, arxiv.org/abs/cs.CC/0603060, (2006).

[29] J. M. M. van Rooij and H. L. Bodlaender, *Design by Measure and Conquer, A Faster Exact Algorithm for Dominating Set*, in Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2008), Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2008, pp. 657-668.

[30] R. E. Tarjan and A. E. Trojanowski, *Finding a maximum independent set*, SIAM Journal on Computing, 6 (1977), pp. 537–546.

[31] B. van der Waerden, *Problem 45*, Jahresber. Deutsch. Math.-Verein., 35 (1926), p. 117.

[32] R. Williams, *A new algorithm for optimal 2-constraint satisfaction and its implications*, Theoretical Computer Science, 348 (2005), pp. 357–365.

[33] G. Woeginger, *Exact algorithms for NP-hard problems: A survey*, in Combinatorial Optimization - Eureka, you shrink!, vol. 2570 of Lecture Notes in Comput. Sci., Springer-Verlag, Berlin, 2003, pp. 185–207.

Appendix

Figure 1 C-like pseudo-code to check a value of λ for given (feasible) values of the weights α's and β's: the function returns `true` if the righthand sides of Recurrences (15)–(20) are upper bounded by λ^k, and `false` otherwise. This is trivially true for Recurrences (13)–(14).

```
boolean feasible(α_1, α_2, α_3, α_4, β_2, β_3, β_4, λ){
  α_0 = β_0 = β_1 = 0; β_5 = β_6 = 1; α_5 = α_4;
  compute Δβ_i for 2 ≤ i ≤ 6 and Δβ_{≤i} for 3 ≤ i ≤ 6;  compute Δα_j and Δα_{≤j} for 2 ≤ j ≤ 5;
  for(r = 2...5) //Case 1
    if(r λ^{-r α_1 - β_r} > 1) return false;
  for(r = 2...5) //Case 2.a
    for(r' = 1...r)
      if(λ^{-α_r - α_{r'} - r β_2} + λ^{-α_r - α_{r'} - r' β_2} > 1) return false;
  for(r = 2...5) //Case 2.b
    for(r_2 = 1...r)
      for(r_3 = 0...r - r_2)
        for(r_4 = 0...r - r_2 - r_3)
          for(r_5 = 0...r - r_2 - r_3 - r_4)
            r_6 = r - r_2 - r_3 - r_4 - r_5;
            if(λ^{-α_r - Σ_{i=2}^{6} r_i β_i - Σ_{i=2}^{6} r_i (i-1) Δα_{≤r}} + λ^{-α_r - r_2 α_2 - β_2 - Σ_{i=2}^{5} r_i Δβ_i} > 1) return false;
  for(r = 3...5) //Case 3
    for(r_3 = 0...r)
      for(r_4 = 0...r - r_3)
        for(r_5 = 0...r - r_3 - r_4)
          r_6 = r - r_3 - r_4 - r_5;
          if(λ^{-α_r - Σ_{i=3}^{6} r_i β_i - Σ_{i=3}^{6} r_i (i-1) Δα_{≤r}} + λ^{-α_r - Σ_{i=3}^{5} r_i Δβ_i} > 1) return false;
  for(r' = 1...2) //Case 4
    for(u' = 3...6)
      for(v' = 3...6)
        if(λ^{-α_2 - α_{r'} - β_{u'} - β_{v'} - (u'+v'-2-r') Δα_{≤2}} + λ^{-α_2 - Δβ_{u'} - Δβ_{v'}} > 1) return false;
  for(u' = 3...6) //Case 5
    for(v' = 3...u')
      if(λ^{-u' α_2 - β_{u'} - β_{v'} - (v'-1) Δα_2 - (u'-1) Δβ_{≤u'}}
            + λ^{-v' α_2 - β_{u'} - β_{v'} - (u'-1) Δα_2 - (v'-1) Δβ_{≤u'}} + λ^{-α_2 - Δβ_{u'} - Δβ_{v'}} > 1) return false;
  return true; //All the inequalities satisfied
}
```