Solving Capacitated Dominating Set by Using Covering by Subsets and Maximum Matching*

Mathieu Liedloff[1], Ioan Todinca[1], and Yngve Villanger[2]

[1] LIFO, Université d'Orléans, BP 6759, F-45067 Orléans Cedex 2, France
{mathieu.liedloff,ioan.todinca}@univ-orleans.fr
[2] Department of Informatics, University of Bergen, N-5020 Bergen, Norway
yngve.villanger@uib.no

Abstract. The Capacitated Dominating Set problem is the problem of finding a dominating set of minimum cardinality where each vertex has been assigned a bound on the number of vertices it has capacity to dominate. Cygan et al. showed in 2009 that this problem can be solved in $O(n^3 m\binom{n}{n/3})$ or in $O^*(1.89^n)$ time using maximum matching algorithm. An alternative way to solve this problem is to use dynamic programming over subsets. By exploiting structural properties of instances that can not be solved fast by the maximum matching approach, and "hiding" additional cost related to considering subsets of large cardinality in the dynamic programming, an improved algorithm is obtained. We show that the Capacitated Dominating Set problem can be solved in $O^*(1.8463^n)$ time.

1 Introduction

The problem of finding a vertex subset of cardinality at most k that dominates all remaining vertices in a graph has received a considerable amount of attention over the last two decades. This problem is known as the Dominating Set problem and is a classical NP-complete and also $W[2]$-complete problem [5]. If each vertex is equipped with a bound on the number of neighbors that it has capacity to dominate additionally to itself, the problem is called Capacitated Dominating Set. By simply assigning the degree as the capacity to each vertex, the hardness results carry over to the Capacitated Dominating Set problem. Recently it has been proven that this problem remains $W[1]$-hard even in the planar case [2], and thus distinguishing it from Dominating Set which is Fixed Parameter Tractable on planar graphs [1].

For the general domination problem there has been a long sequence of moderately exponential time algorithms starting in 2004 by [8], where the currently last result is [10]. Also in the case where we ask for a connected dominating set the trivial bound was broken back in 2006 [6].

* This work has been supported by ANR Blanc AGAPE (ANR-09-BLAN-0159-03) and EGIDE Aurora (18809RM).

D.M. Thilikos (Ed.): WG 2010, LNCS 6410, pp. 88–99, 2010.

The Capacitated Dominating Set problem have a slightly different story. The question if the “trivial” $O^*(2^n)$ could be broken was first asked at IWPEC 2008, and repeated later that year by Johan van Rooij at Dagstuhl [7]. Cygan et al. show in [4] that the Capacitated Dominating Set problem could be solved exactly in $O^*(1.89^n)$ time. They first provide an algorithm that, given a set U of vertices, computes in polynomial time a minimum dominating set D with $U \subseteq D$ and such that only the elements of U are allowed to dominate two or more vertices outside D. This algorithm is based on a reduction to a maximum matching problem. Then the result is obtained by guessing U, which is of size no more than $n/3$.

Koivisto [9] gave a clever algorithm for the following problem: the input is a family $\mathcal{F}$ of subsets of a universe V and the output is a partition of V into a minimum number of elements of $\mathcal{F}$. The running time is $o(2^n)$ if all the sets of $\mathcal{F}$ are small (bounded by a constant). This yields an $o(2^n)$ algorithm for the Capacitated Dominating problem when all capacities are all upper bounded by a constant k. Indeed, we can put in $\mathcal{F}$ all subsets $X \subseteq N[v]$ such that $|X| \leq c(v)+1$, for all vertices v (see next section for details).

For both Cygan's et al. and Koivisto's algorithms there exist instances that forces the algorithms to spend respectively $O^*(1.89^n)$ and $O^*(2^n)$ time. There are even instances that force both algorithms to their respective maximum bound. Consider a Capacitated Dominating Set D where a constant number of vertices W dominate ρn vertices for a constant $0 < \rho < 1$, and every vertex of $D \setminus W$ uses its capacity to dominate exactly two vertices in $V \setminus D$. Such an instance will force Koivisto's algorithm to consider dominating vertices that use their capacity to dominate ρn other vertices (hence the sets of $\mathcal{F}$ are not of bounded size), and the maximum matching approach by Cygan et al. has to test all subsets U of size $|W| + (n - |W|\rho n)/3$. Thus, no significant improvement can be obtained by simply balancing the two approaches.

In this paper we adapt the algorithm by Koivisto to avoid the “constant” restriction on the capacities, meaning that the elements of the family $\mathcal{F}$ will not necessarily be of bounded size. This generalization works under the condition that there are not too many vertices in the dominating set using unbounded capacity. Maybe the most interesting contribution here is that handling elements of $\mathcal{F}$ of unbounded size does not add to the total running time of the algorithm. The reason for this comes form the fact that all unbounded sets are handled in the beginning of the dynamic programming, before the considering smaller sets. This is then balanced with the maximum matching approach of Cygan et al. to optimize the time bound. As a result of this we get that the Capacitated Dominating Set problem can be solved in $O^*(1.8463^n)$ time. When we look further into the polynomial factors hidden by the big-Oh notation, we can actually notice that there is a trade-off between the polynomial factor and the basis of the exponent, with running times ranging from $O(n^9 \cdot 1.8844^n)$ to $O(n^{40005} \cdot 1.8463^n)$.

2 Preliminaries

In this paper we consider simple and undirected graphs. Given a graph $G = (V, E)$, we denote by n the number of its vertices. If the given graph is equipped with a capacity function $c : V \to \mathbb{N}$, we say that a subset $D \subseteq V$ is a Capacitated Dominating Set if there exists a function $f : V \setminus D \to D$ such that $f(u) \in N(u) \cap D$ and $|f^{-1}(v)| \leq c(v)$ for each $v \in D$. Alternatively the CAPACITATED DOMINATING SET problem can be viewed as a partitioning problem. Let $\mathcal{F}$ be the family of subsets of V such that $X \subseteq V$ is a member of $\mathcal{F}$ if and only if there exists a vertex $v \in X$ such that $X \subseteq N[v]$ and $|X| \leq c(v) + 1$. Such a vertex v will be refereed to as the *representative* of X and is denoted $v = R(X)$. Any partitioning $S_1, S_2, \ldots, S_d$ of V where $S_i \in \mathcal{F}$ for every $i \in \{1, \ldots, d\}$ defines a Capacitated Dominating Set D of G, where $|D| = d$. The capacitated dominating set D can be retrieved from the partitioning by selecting the representative of each set, i.e. $D = \cup_{i=1}^{d} R(S_i)$. In the next section, we will view the CAPACITATED DOMINATING SET problem as the problem of finding such a partitioning where d is minimized.

3 The Two Main Ingredients

We start by briefly recalling the construction of Cygan et al. and Koivisto's algorithms.

3.1 Cygan et al.'s Algorithm

Let $G = (V, E)$ be a graph with a capacity function c. In [4], Cygan, Pilipczuk and Wojtaszczyk give a $O^*(1.89^n)$ time exact algorithm to compute a capacitated dominating set. This algorithm heavily relies on a reduction to a matching problem. Namely, they consider the Constrained CDS problem defined as follows: Given a set U of representatives of sets of size at least 3, compute a smallest CDS $D \subseteq V$ such that $U \subseteq D$, and $D \setminus U$ are the representatives of sets of size at most 2. In words, the vertices of $D \setminus U$ will dominate at most one vertex outside D. Cygan et al. give a polynomial algorithm for this problem. We refer to this algorithm as `ExtendSolution(`G, c, U`)`. By enumerating all subsets U of size at most $n/3$, one can solve the CDS problem in $O^*(\binom{n}{n/3}) = O^*(1.89^n)$ time.

Theorem 1 (see [4]). *Given a set U of representatives of sets of size at least 3, algorithm* `ExtendSolution(`G, c, U`)` *computes a smallest CDS $D \subseteq V$ such that $U \subseteq D$, in $O(n^2 m)$ running time. Consequeltly, the CDS problem can be solved in $O^*(1.89^n)$ time.*

3.2 Koivisto's Algorithm: Partitioning into Sets

In [9], Koivisto investigates the problem of finding a partition of a universe V into k disjoints sets $S_1, S_2, \ldots, S_k$ from a family $\mathcal{F}$ of subsets of V under the restriction that the subsets of $\mathcal{F}$ are of bounded cardinality. Thanks to an (arbitrary) linear order $<$ on the elements of V (implying a lexicographic order $\prec$ on the subsets of V), Koivisto designs an algorithm with a proved worst-case running-time $O^*(c^n)$ with $c < 2$. We recall briefly the approach developped by Koivisto. In the next section, we show how to extend this approach if the familly $\mathcal{F}$ contains *some* sets of *unbounded* cardinality. Formally, the following was proved in [9]:

Theorem 2. *Given an n-element universe V, a number k, and a familly $\mathcal{F}$ of subsets of V, each of cardinality at most r, the partitions of V into k members of $\mathcal{F}$ can be counted in time $O^*(|\mathcal{F}|2^{n\lambda_r})$ where $\lambda_r = \frac{2r-2}{\sqrt{(2r-1)^2 - 2\ln 2}}$.*

To achieve this result, one idea of Koivisto approach is to define an arbitrary linear order $<$ on the universe and to count the lexicographic ordered k-partitions $(S_1, S_2, \ldots, S_k)$ of V such that $S_i \prec S_j$ whenever $1 \leq i < j \leq k$. It is shown in [9] that w.l.o.g the number of lexicographic ordered k-partition is equal to the number of k-partitions.

The algorithm is very simple and is based on dynamic programming: for any $W \subseteq V$ and integer j, $1 \leq j \leq k$, let $f_j(W)$ be the number of ordered partitions of W into j sets of $\mathcal{F}$. Clearly, we have (see [9] for details):

$$f_1(W) = [W \in \mathcal{F}] \quad \text{and} \quad f_j(W) = \sum_{X \subseteq W} f_{j-1}(W \setminus X)[X \in \mathcal{F}] \quad \text{for } j > 1.$$

Here $[W \in \mathcal{F}]$ counts the occurences of the set W in the family $\mathcal{F}$.

As observed in [9], considering only lexicographic ordered partitions leads to a reduction in the number of subsets of V needed to be considered by the algorithm. Indeed, the set S_j must contain the smallest element of V not in $S_1 \cup S_2 \cup ... \cup S_{j-1}$. Let $\mathcal{R}_j$ be the familly of sets W that is the union of j such sets $S_1, S_2, \ldots, S_j$. This familly is defined recursively by:

$$\mathcal{R}_1 = \{X \text{ s.t. } X \in \mathcal{F},\ \min V \in X\};$$
$$\mathcal{R}_j = \{Y \cup X \text{ s.t. } Y \in \mathcal{R}_{j-1},\ X \in \mathcal{F},\ Y \cap X = \emptyset,\ \min V \setminus Y \in X\}.$$

It is shown in [9] that the running-time of the algorithm is proportional to $(|\mathcal{R}_1| + |\mathcal{R}_2| + \cdots + |\mathcal{R}_k|) \cdot |\mathcal{F}|$. Note that if we require the size of each sets in $\mathcal{F}$ to be bounded by a constant r, then the size of the family $\mathcal{F}$ is no more than n^r. Now we derive an upper bound on each $|\mathcal{R}_j|$ being sharper than the one of Theorem 2. However, contrary to Theorem 2 which provides an algebraic expression on the running-time, our result will involve calculus, see the next theorem and Table 1 for some values.

Theorem 3. *Given an n-element universe V, a number k, and a familly $\mathcal{F}$ of subsets of V, each of cardinality at most r, the partitions of V into k members of $\mathcal{F}$ can be counted in time $O^*\left(|\mathcal{F}| \cdot \binom{(1-\lambda)n}{(r-1)\lambda n}\right)$ where λ is the unique solution of $\frac{(1-\lambda r)^r}{(\lambda(r-1))^{r-1}} = 1 - \lambda$ in $[0; \frac{1}{2r-1}]$.*

Proof. Let V an n-element universe with an arbitrary linear order on its elements: $v_1, v_2, \ldots, v_n$. Let j be an integer of $\{1, 2, \ldots, k\}$. Recall that $\mathcal{R}_j$ is the familly of sets W being the union of j disjoint sets $S_1, S_2, \dots, S_j$, each of size no more than r. Also by definition of $\mathcal{R}_j$, for any set $W \in \mathcal{R}_j$, $\{v_1, v_2, \dots, v_j\} \subseteq W$. Thus the size of $\mathcal{R}_j$ is upper bounded by the maximum number of different sets W that it can contain. This number is no more than the largest value of $\binom{n-j}{w-j}$ for $j \le w \le rj$. Indeed, the size of W, denoted here by w is at least j since W contains the j first vertices and at most rj since W is the disjoint union of j sets of size at most r. As W has to contain the j first vertices, the $|W| - j$ others vertices have to be choosen among the $n - j$ remaining vertices.

By denoting ρn, with $0 \le \rho \le 1$, the value of j and by w', with $0 \le w' \le (r-1)\rho$, the value of $w - j$, the expression $\binom{n-j}{w-j}$ can be rewritten $\binom{n(1-\rho)}{nw'}$. This latter expression is maximum for $w' = \frac{1-\rho}{2}$ whenever $\frac{1-\rho}{2} \le (r-1)\rho$ or for $w' = (r-1)\rho$ otherwise. (This can easily be seen from the well-known binomial formula.) Note that $\frac{1-\rho}{2} = (r-1)\rho$ for $\rho = \frac{1}{2r-1}$.

Also, by Stirling's approximation, $\binom{\alpha n}{\beta n}$ is asymptotically bounded by $B(\alpha, \beta)^n$ where $B : (\alpha, \beta) \mapsto \frac{\alpha^\alpha}{\beta^\beta (\alpha-\beta)^{\alpha-\beta}}$.

Thus in the next part of the proof we study the functions $f_1 : \rho \mapsto B(1 - \rho, (r-1)\rho)$ defined over $[0; \frac{1}{2r-1}]$ and $f_2 : \rho \mapsto B(1-\rho, \frac{1-\rho}{2})$ defined over $[\frac{1}{2r-1}; 1]$ (see Fig. 1 for a plot of these two functions).

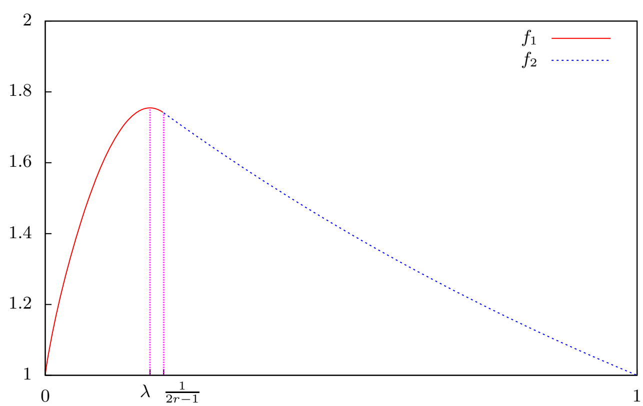

Fig. 1. The two functions f_1 and f_2 plotted for $r = 3$. The maximum of f_1 is reached for λ.

We first start by studying f_2. Its derivative $f_2' = -(1-\rho)^{1-\rho}(\frac{1-\rho}{2})^{\rho-1}\ln(2)$ is non positive on $[\frac{1}{2r-1}; 1]$. Thus the maximum of f_2 is obtained for $\rho = \frac{1}{2r-1}$. Since $f_2(\frac{1}{2r-1}) = f_1(\frac{1}{2r-1})$, it sufficient to restrict ourselves to the analysis of function f_1 over $[0; \frac{1}{2r-1}]$.

Then we consider f_1. Its derivative is $f_1' = (1-\rho)^{1-\rho}((r-1)\rho)^{(1-r)\rho}(1 - \rho r)^{\rho r - 1}(-\ln(1-\rho) - r\ln((r-1)\rho) + \ln((r-1)\rho) + r\ln(1-\rho r))$ and is equal to zero over $[0; \frac{1}{2r-1}]$ if and only if ρ satisfies $(-\ln(1-\rho) - r\ln((r-1)\rho) +$

$\ln((r-1)\rho) + r\ln(1-\rho r)) = 0$. In other words, whenever $\rho \in [0; \frac{1}{2r-1}]$ satifies $\frac{(1-\rho r)^r}{(\rho(r-1))^{r-1}} = 1-\rho$ in $[0; \frac{1}{2r-1}]$. Let us show that such a zero exists and is unique. (The zero will correspond to λ on Fig. 1.) Consider $\tilde{f}_1' : \rho \mapsto -\ln(1-\rho) - r\ln((r-1)\rho) + \ln((r-1)\rho) + r\ln(1-\rho r)$; its derivative is $\tilde{f}_1'' : \frac{1-r}{\rho(\rho-1)(\rho r-1)}$ and is negative over $[0; \frac{1}{2r-1}]$ for any $r > 1$. Thus $\tilde{f}_1'$ is strictly decreasing over $[0; \frac{1}{2r-1}]$ and admits a unique zero since $\tilde{f}_1'(0) \geq 0$ and $\tilde{f}_1'(\frac{1}{2r-1}) \leq 0$. □

Now we shortly explain how to compute a *good* approximation of λ (defined as in Theorem 3) and then provide an asymptotic upper bound on the running-time claimed by Theorem 3.

We have shown that f_1' has a unique zero over $[0; \frac{1}{2r-1}]$; it can easily be shown that f_1' is decreasing and monotone over $[0; \frac{1}{2r-1}]$ with f_1' positive over $[0; \lambda]$ and negative over $[\lambda; \frac{1}{2r-1}]$. Here, λ is the unique solution of $\frac{(1-\lambda r)^r}{(\lambda(r-1))^{r-1}} = 1-\lambda$ in $[0; \frac{1}{2r-1}]$.

Let ϵ be a positive real. Let $\tilde{\lambda}$ "close enough to λ", i.e. so that $f_1'(\tilde{\lambda}-\epsilon) \geq 0$ and $f_1'(\tilde{\lambda}+\epsilon) \leq 0$. Since the zero of f_1' is unique, such a $\tilde{\lambda}$ can be found by binary search. Since f_1' is monotone, it follows that $\tilde{\lambda}-\epsilon > \lambda > \tilde{\lambda}+\epsilon$ and thus $|\tilde{\lambda}-\lambda| \leq \epsilon$. Let $\Delta = \max\left(|f_1'(\tilde{\lambda}-\epsilon)|, |f_1'(\tilde{\lambda}+\epsilon)|\right)$. By the classical *mean value* theorem, $|f(\tilde{\lambda}) - f(\lambda)| \leq \Delta \cdot |\tilde{\lambda}-\lambda| \leq \Delta \cdot \epsilon$. Consequently we have $f(\lambda) \leq f(\tilde{\lambda}) + \Delta \cdot \epsilon$. Finally we recall that $O^*(f(\lambda)^n) = \binom{(1-\lambda)n}{(r-1)\lambda n}$.

Thus it is sufficient to find a $\tilde{\lambda}$ close enough to λ by binary search to establish an upper bound on the running-time of Koivisto algorithm. Table 1 provides values of $\tilde{\lambda}$ for some values of r. We also give the corresponding running-time obtained by Theorem 3 (named as "Koivisto*") and the running-time from the original analysis by Koivisto [9] (see column "Koivisto").

Table 1. The table provides, for several values of r, the running-time of Koivisto's algorithm [9] according to Koivisto's analysis (column "Koivisto") and according to the new analysis devised by Theorem 3 (column "Koivisto*"). We also provide the value of $\tilde{\lambda}$ which act as a certificate together with $\epsilon = 10^{-5}$ for proving the correctness of values given by Koivisto*. Note that for the special case $r = 2$, the analysis provided by [9] is already sharp.

r	Koivisto	**Koivisto***	$\tilde{\lambda}$
2	1.6181	**1.6181**	0.27629
3	1.7693	**1.7549**	0.17701
4	1.8271	**1.8192**	0.13051
10	1.9308	**1.9296**	0.05081
20	1.9654	**1.9651**	0.02520
50	1.9862	**1.9861**	0.01003
100	1.9931	**1.9931**	0.00501

3.3 A New Recipe for Capacitated Dominating Set: Combining and Cooking the Ingredients

We are now ready to explain how the Capacitated Dominating Set problem can be solved in time $O^*(1.8463^n)$ by combining the approach of Cygan et al. [4] and an extended version of the result by Koivisto [9] for bounded cardinalty sets family. Namely, we will show that Koivisto's approach can be used even if *some* sets of the family are of unbounded size. To do that, instead of fixing an arbitrary ordering over the elements of a universe, we use this freedom to put some elements playing a special role at the begining of the ordering. These elements are the representatives of sets being of unbounded cardinality (i.e. not constant bounded). As we will see, in our Capacitated Dominating Set algorithm, these elements have to be guessed. It is worth to note that if these elements are given as part of the input, the running-time analysis can be improved. Then, once the unbounded sets have been guessed, the algorithm has to deal only with bounded-cardinality sets.

Description of the Algorithm. Let $G = (V, E)$ be a graph with a capacity function c defined over V. Let $\mathcal{F}$ be the family of subsets $X \subseteq V$ such that $X \in \mathcal{F}$ if and only if there exists a vertex $v \in X$ such that $X \subseteq N[v]$ and $|X| \leq c(v) + 1$. We recall here that such a $v = R(X)$ is called the representative of X. Note that it is possible that two sets X_1 and X_2 with $X_1 = X_2$, $R(X_1) = u$, $R(X_2) = v$, $u \neq v$ satisfy the required properties for belonging to $\mathcal{F}$. In that case, it is sufficient to keep only one of these sets in $\mathcal{F}$ together with its corresponding representative. In that way, given a $X \in \mathcal{F}$, its representative is unique. From now on a family $\mathcal{F}$ constructed as explained is called a *family*; we will omit to precise the universe V and the capacity function c when it is clear from the context.

Lemma 1. *Let $\mathcal{F}$ be a family. If $S_1, S_2, \ldots, S_d$, with $S_i \in \mathcal{F}$ for each $i \in \{1, \ldots, d\}$, is a partition of the universe V then their representatives are pairwise distinct.*

Proof. Since $S_1, S_2, \ldots, S_d$ is a partition of V and for each $i \in \{1, \ldots, d\}$, $R(S_i) \in S_i$, the lemma follows. □

Let $\beta \in \mathbb{N}$ with $3 < \beta \leq n$. Let $\mathcal{F}$ be a family and let $S_1, S_2, \ldots, S_d$ be a partition of V. We enrich the notion of representatives with the notions of *big representatives*, *medium representatives* and *tiny representatives* with respect to this partition. We say that a vertex v is a

- *big representative* if there is a S_i, $1 \leq i \leq d$, with $v = R(S_i)$ and $|S_i| \geq \beta$;
- *medium representative* if there is a S_i, $1 \leq i \leq d$, with $v = R(S_i)$ and $\beta > |S_i| \geq 3$;
- *tiny representative* if there is a S_i, $1 \leq i \leq d$, with $v = R(S_i)$ and $3 > |S_i|$;

The set of all big (resp. medium and tiny) representative is denoted by `(BR)` (resp. `(MR)` and `(TR)`).

To simplify further the description, given a disjoint collection $S_1, S_2, \ldots, S_d$ of subsets and a representative v, we denote by $S(v)$ the set S_i such that $v = R(S_i)$.

How does the algorithm work? A complete and formal description of the algorithm is given by Algorithm `minCDS`. Suppose that $D \subseteq V$ is a solution of size d of the minimum Capacitated Dominating Set problem. Then, there exists a partition of V into d sets $S_1, S_2, \dots, S_d$ such that each $v \in D$ is the representative of one set S_i with $|S_i| \leq c(v) + 1$. Thus D can also be partitioned into (BR), (MR) and (TR).

First we start in "Step 1" (see Algorithm `minCDS`) by computing all possible solutions (possibly not of minimum size) such that $|\texttt{(BR)} \cup \texttt{(MR)}| \leq \gamma n$, for some fixed $\gamma \in [1/4; 1/3]$. This is done using Algorithm `ExtendSolution`. Then in all solutions computed in "Step 2" we may assume that $|\texttt{(BR)} \cup \texttt{(MR)}| \geq \gamma n$. Since all sets $S(v)$ with $v \in \texttt{(BR)} \cup \texttt{(MR)}$ are of cardinality at least 3, it follows that (BR) is of moderate size as shown by the next lemma. This set (BR) of big representatives is guessed by our algorithm.

Lemma 2. *Suppose that there exists a solution $S_1, S_2, \dots, S_d$ such that $|\texttt{(BR)} \cup \texttt{(MR)}| \geq \gamma n$, for $\gamma \in [0; 1]$. Then the size of* (BR) *is at most $\frac{n-3\gamma n}{\beta-3}$.*

Proof. Each set S_i such that $R(S_i) \in \texttt{(BR)} \cup \texttt{(MR)}$ has cardinality at least 3. Thus only $n - 3\gamma n$ vertices can be distributed over the S_i's having a big representative. Since sets with a big representative are of size at least β (and already contain 3 vertices), it follows that $|\texttt{(BR)}| \leq \frac{n-3\gamma n}{\beta-3}$. □

In addition, for each S_i with a big representative, the size of S_i is not necessary bounded by a constant, but a linear bound can be established on its size:

Lemma 3. *Suppose that there exists a solution $S_1, S_2, \dots, S_d$ such that $|\texttt{(BR)} \cup \texttt{(MR)}| \geq \gamma n$, for $\gamma \in [0; 1]$. Then the size of each S_i such that $R(S_i) \in$ (BR) is at most $n - 3\gamma n + 3$. Moreover, the following is satisfied $|\cup_{v \in \texttt{(BR)}} S(v)| \leq \frac{n\beta - 3\gamma\beta n}{\beta-3}$.*

Proof. As shown in the proof of Lemma 2, only $n - 3\gamma n$ vertices can be distributed over the S_i having a big representative, assuming that each S_j with $R(S_j) \in \texttt{(BR)} \cup \texttt{(MR)}$ has cardinality at least 3. Thus, each such S_i has at most $n - 3\gamma n + 3$ vertices. Since by Lemma 2, the number of sets with a big representative is at most $\frac{n-3\gamma n}{\beta-3}$, it follows that $|\cup_{v \in (BR)} S(v)| \leq 3 \cdot \frac{n-3\gamma n}{\beta-3} + n - 3\gamma n$. □

In "Step 2.1", our algorithm deals with such sets S_i having a big representative. These sets can only appear during this step since we put the big representatives (guessed by the foreach-loop) at the very beginning of ordering $<$. Then we start to compute partitions of V into sets using the dynamic programming approach recalled in Section 3.2 (see also [9]). Here the size of the sets of $\tilde{\mathcal{F}}$ are not bounded by a constant; nevertheless we will show in Section 3.3 that a *good* bound on the running-time can be established. Finally, in "Step 2.2" it only remain sets of size bounded by the constant $\beta - 1$, and the dynamic programming is pursued. We combine the possible cases in order to retrieve the global optimum solution.

Remark 1. It is straightforward to adapt our Algorithm `minCDS` so that it returns a minimum capacitated dominating set instead of its size.

```
Algorithm minCDS(G = (V, E), c : V → ℕ)
Input: A graph G = (V, E) and a capacity function c.
Output: The size of a minimum Capacitated Dominating Set of G.
  /* γ is a constant that has to be choosen in [1/4; 1/3] and β is a
     constant so that β > 3 (see Section 3.3)                          */
  γ ← 31/100
  β ← 15
  MinSol ← ∞

  /* --- Step 1 : based on Cygan et al. approach ---                   */
  for ℓ = 0 to γn do
      foreach U ⊆ V of size ℓ do
          MinSol ← min{MinSol, ℓ + ExtendSolution(G, c, U)}

  /* --- Step 2 : based on Koivisto approach ---                       */
  for ℓ = 0 to (n−3γn)/(β−3) do
      foreach (BR) ⊆ V of size ℓ do
          Define an ordering < by putting first the vertices of (BR) (in arbitrary
          order) and then the vertices of V \ (BR) (in arbitrary order)
          forall i ∈ {0, 1, ..., n} do R_i ← ∅
          /* -- Step 2.1: dealing with sets of size ≥ β --              */
          z ← n − 3γn
          Let
          F̃ = {X ⊆ V, |X| ≤ z + 3, ∃v ∈ (BR) s.t. X ⊆ N[v] and |X| ≤ c(v) + 1}
          for i = 1 to ℓ do
              foreach Y ∈ R_{i−1} and X ∈ F̃ s.t. Y ∩ X = ∅ and min V \ Y ∈ X do
                  Add Y ∪ X to R_i

          /* -- Step 2.2: dealing with sets of size < β --              */
          Let
          F = {X ⊆ V, |X| < β, ∃v ∈ V \ (BR) s.t. X ⊆ N[v] and |X| ≤ c(v) + 1}
          for i = ℓ + 1 to n do
              foreach Y ∈ R_{i−1} and X ∈ F s.t. Y ∩ X = ∅ and min V \ Y ∈ X do
                  Add Y ∪ X to R_i

          Let i be the smallest index such that V ∈ R_i
          MinSol ← min{MinSol, i}
  return MinSol
```

Running-Time Analysis. In this section we show that the worst-case running-time of Algorithm `minCDS` is $O^*(1.8573^n)$ (using $\gamma = 31/100$ and $\beta = 15$ as stated in the algorithm). With some appropriate values for γ and β which are used as constants by Algorithm `minCDS`, this worst-case running-time can be lowered to $O^*(1.8463^n)$. We already emphasys that a big polynomial is hidden in this lattest big-Oh notation. This issue will be discussed in Section 3.3 (see also Table 2).

Lemma 4. *The running-time of Step 1 is bounded by* $O(n^3 m \binom{n}{\gamma n})$.

Proof. The total number of sets U is $\sum_{\ell=1}^{\gamma n} \binom{n}{\ell} \leq n\binom{n}{n/3}$ since $\gamma \leq n/3$. Each call to `ExtendSolution` costs $O(n^2 m)$ time by Theorem 1. Total time for this step becomes $O(n^3 m\binom{n}{\gamma n})$. □

By the same argument as above combined with Lemma 2 and Lemma 3 we get:

Lemma 5. *The number of sets* (BR) *considered by the outmost ForEach-loop in Step 2 is at most* $O\left(n\binom{n}{\frac{n-3\gamma n}{\beta-3}}\right)$.

Lemma 6. *The size of the family* $\tilde{\mathcal{F}}$ *is bounded by* $O(n\binom{n}{n-3\gamma n+3})$.

Lemma 7. *Let* $1/4 < \gamma$. *The running-time of Step 2.1 is bounded by* $O\big(n^4|\tilde{\mathcal{F}}| \cdot \binom{(1-\lambda)n}{(2\lambda+1-3\gamma)n}\big)$ *where* λ *is the (possible) unique real solution of* $27(\gamma-\lambda)^3 = (1-\lambda)(1-3\gamma+2\lambda)^2$ *over* $[0; \frac{1-3\gamma}{\beta-3}]$ *if such a solution exists, or the running-time is bounded by* $O\big(n^4|\tilde{\mathcal{F}}| \cdot \binom{(\frac{\beta-4+3\gamma}{\beta-3})n}{(\frac{1+3\gamma+\beta-3\gamma\beta}{\beta-3})n}\big)$ *otherwise.*

Proof. We first provide a bound on the size of the family $\mathcal{R}_i$, $1 \leq i \leq \ell$, where ℓ denotes the size of (BR). For each $W \in \mathcal{R}_i$ its size is at most $3i + n - 3\gamma n$ (recall that W is the union of at most i sets S of cardinality at least 3; furthermore at most $n-3\gamma n$ vertices can be distributed over all these sets — see Lemma 3 and its proof). However, due to the ordering $<$ and by the construction of $W \in \mathcal{R}_i$, each $W \in \mathcal{R}_i$ has to contains the first i elements. Thus there are at most $n\binom{n-i}{2i+n-3\gamma n}$ possible W in $\mathcal{R}_i$.

By Stirling's approximation, $\binom{\alpha n}{\beta n}$ is asymptotically bounded by $B(\alpha,\beta)^n$ where $B : (\alpha,\beta) \mapsto \frac{\alpha^\alpha}{\beta^\beta(\alpha-\beta)^{\alpha-\beta}}$. As done in the proof of Theorem 3, consider the function $f_1 : \rho \mapsto B(1-\rho, 2\rho+1-3\gamma)$ defined over $[0; \frac{1-3\gamma}{\beta-3}]$. Its derivative is equal to zero if and only if $3\ln(3) + 3\ln(\gamma-\rho-\ln(1-\rho) - 2\ln(1-3\gamma+2\rho) = 0$, or in other words whenever $27(\gamma-\rho)^3 = (1-\rho)(1-3\gamma+2\rho)^2$. By standard calculation, this equation has a unique solution, if it exists, over $[0; \frac{1-3\gamma}{\beta-3}]$. Otherwise, if no solution exists, then f_1 is increasing over $[0; \frac{1-3\gamma}{\beta-3}]$ and its maximum is $f_1(\frac{1-3\gamma}{\beta-3})$. For the polynomial contribution of the running time we get n^2 for running from 0 to ℓ and then 0 to n, each set $\mathcal{R}_i$ might contain subsets of $O(n)$ different sizes, and finaly we need $O(n)$ time to find sets and check presence of edges. □

Lemma 8. *Let* $\gamma \in [0.18995; 1/3]$. *The running-time of Step 2.2 is bounded by* $O\big(n^4 \cdot n^\beta \cdot \binom{(1-\lambda)n}{(1-3\gamma+2\lambda)n}\big)$ *where* λ *is the unique real root of* $(1-\lambda)(1-3\gamma+2\lambda)^2 - 27(\gamma-\lambda)^3$ *over* $[0; \frac{6\gamma-1}{5}]$.

Proof. The proof is quite similar to the previous one. Again each $W \in \mathcal{R}_i$, $\ell < i \leq n$, is of size at most $3i + n - 3\gamma n$. Since it is required that each such set W contains the first i elements, it follows that the size of $\mathcal{R}_i$ is at most $\binom{n-i}{2i+n-3\gamma n}$.

Table 2. The table provides worst-case running-times of Algorithm `minCDS`, depending on the values for γ and β. The order of the hidden polynomial term in the big-Oh notation is given by the second column.

Running-Time	order of the polynomial	γ	β
$O^*(1.8844^n)$	$n^5 \cdot n^4$	0.32914	4
$O^*(1.8798^n)$	$n^5 \cdot n^5$	0.32574	5
$O^*(1.8649^n)$	$n^5 \cdot n^{10}$	0.31520	10
$O^*(1.8573^n)$	$n^5 \cdot n^{15}$	0.31000	15
$O^*(1.8486^n)$	$n^5 \cdot n^{50}$	0.30424	50
$O^*(1.8463^n)$	$n^5 \cdot n^{40000}$	0.30275	40000

Now we consider the functions $f_1 : \rho \mapsto B(1-\rho, 1-3\gamma+2\rho)$ defined over $[0; \frac{6\gamma-1}{5}]$ and $f_2 : \rho \mapsto B(1-\rho, \frac{1-\rho}{2})$ defined over $[\frac{6\gamma-1}{5}; 1]$. (To justify this cut between f_1 and f_2, observe that $(1-\rho)/2 \leq 1-3\gamma+2\rho$ whenever $\rho \geq \frac{6\gamma-1}{5}$.) It can easily be shown that f_2 is decreasing over $[\frac{6\gamma-1}{5}; 1]$ and thus we can restrict ourself on f_1. Again, by studying its derivative, we claim that f_1 is maximum over $[0; \frac{6\gamma-1}{5}]$ for λ being the unique real root of $(1-\lambda)(1-3\gamma+2\lambda)^2-27(\gamma-\lambda)^3$. By the same agruments as used in the proof of Lemma 7, we get the polynomial factor to be n^4. □

By combining the previous lemmata, we establish the following bound on the worst-case running-time:

Theorem 4. *The worst-case running-time of Algorithm* `minCDS` *is the maximum over:*

- *Step 1 :* $O\big(n\binom{n}{\gamma n}\big)$ *(by Lemma 4);*
- *Step 2.1 :* $O\big(n^6\binom{n}{\frac{n-3\gamma n}{\beta-3}}\cdot\binom{n}{n-3\gamma n+3}\cdot\binom{(1-\lambda)n}{(2\lambda+1-3\gamma)n}\big)$ *if a solution* λ *of* $27(\gamma-\lambda)^3=(1-\lambda)(1-3\gamma+2\lambda)^2$ *exists over* $[0; \frac{1-3\gamma}{\beta-3}]$ *; otherwise* $O^*\big(n^6\binom{n}{\frac{n-3\gamma n}{\beta-3}}\cdot\binom{n}{n-3\gamma n+3}\cdot\binom{(\frac{\beta-4+3\gamma}{\beta-3})n}{(\frac{1+3\gamma+\beta-3\gamma\beta}{\beta-3})n}\big)$ *(by Lemma 5, Lemma 6 and Lemma 7);*
- *Step 2.2 :* $O\big(n^5\binom{n}{\frac{n-3\gamma n}{\beta-3}}\cdot\binom{(1-\lambda)n}{(1-3\gamma+2\lambda)n}\big)$ *where* λ *is the unique real root of* $(1-\lambda)(1-3\gamma+2\lambda)^2-27(\gamma-\lambda)^3$ *over* $[0; \frac{6\gamma-1}{5}]$ *(by Lemma 5 and Lemma 8).*

We finally derive to the following corollary:

Corollary 1. *By setting* $\gamma = 31/100$ *and* $\beta = 15$*, Algorithm* `minCDS` *runs in* $O(n^{20}1.8573^n)$ *and exponential space.*

A Trade-Off between Polynomial and Exponential Terms. As shown in Theorem 4, the running-time of Algorithm `minCDS` depends on two parameters: γ and β. The parameter β has a direct influence on the order of the polynomial term which appears in the running-time. As well, we recall that the size of the family $\mathcal{F}$ of subsets also contributes to the running-time of the algorithm in [9]. Thus, by adequately tuning the parameters (i.e. with $\gamma = 0.30275$ and $\beta = 40000$) Theorem 4 shows that the algorithm runs in $O^*(1.8463^n)$. However the big-Oh notation hides a *huge* polynomial term of order n^β. In Table 2 we give some possible running-times achieve by our algorithm for several values of γ and β.

References

1. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica 33(4), 461–493 (2002)
2. Bodlaender, H.L., Lokshtanov, D., Penninkx, E.: Planar capacitated dominating set is $W[1]$-Hard. In: Chen and Fomin [3], pp. 50–60
3. Chen, J., Fomin, F.V. (eds.): 4th International Workshop on Parameterized and Exact Computation, IWPEC 2009, Copenhagen, Denmark, Revised Selected Papers, September 10-11. LNCS, vol. 5917. Springer, Heidelberg (2009)
4. Cygan, M., Pilipczuk, M., Wojtaszczyk, J.O.: Capacitated domination faster than $O(2^n)$. In: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 74–80. Springer, Heidelberg (2010)
5. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Heidelberg (1999)
6. Fomin, F.V., Grandoni, F., Kratsch, D.: Solving connected dominating set faster than 2^n. In: Arun-Kumar, S., Garg, N. (eds.) FSTTCS 2006. LNCS, vol. 4337, pp. 152–163. Springer, Heidelberg (2006)
7. Fomin, F.V., Iwama, K., Kratsch, D., Kaski, P., Koivisto, M., Kowalik, L., Okamoto, Y., van Rooij, J., Williams, R.: 08431 open problems moderately exponential time algorithms. In: Fomin, F.V., Iwama, K., Kratsch, D. (eds.) Moderately Exponential Time Algorithms. No. 08431 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany (2008), `http://drops.dagstuhl.de/opus/volltexte/2008/1798`
8. Fomin, F.V., Kratsch, D., Woeginger, G.J.: Exact (exponential) algorithms for the dominating set problem. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) WG 2004. LNCS, vol. 3353, pp. 245–256. Springer, Heidelberg (2004) c
9. Koivisto, M.: Partitioning into sets of bounded cardinality. In: Chen and Fomin [3], pp. 258–263
10. van Rooij, J.M.M., Nederlof, J., van Dijk, T.C.: Inclusion/exclusion meets measure and conquer. In: Fiat, A., Sanders, P. (eds.) ESA 2009. LNCS, vol. 5757, pp. 554–565. Springer, Heidelberg (2009)