Parameterized Domination in Circle Graphs*

Nicolas Bousquet[1], Daniel Gonçalves[1], George B. Mertzios[2], Christophe Paul[1], Ignasi Sau[1], and Stéphan Thomassé[3]

[1] AlGCo Project-Team, CNRS, LIRMM, Montpellier, France
{FirstName.FamilyName}@lirmm.fr
[2] School of Engineering and Computing Sciences, Durham University, U.K.
george.mertzios@durham.ac.uk
[3] Laboratoire LIP (U. Lyon, CNRS, ENS Lyon, INRIA, UCBL), Lyon, France
stephan.thomasse@ens-lyon.fr

Abstract. A *circle graph* is the intersection graph of a set of chords in a circle. Keil [*Discrete Applied Mathematics*, 42(1):51-63, 1993] proved that Dominating Set, Connected Dominating Set, and Total Dominating Set are NP-complete in circle graphs. To the best of our knowledge, nothing was known about the parameterized complexity of these problems in circle graphs. In this paper we prove the following results, which contribute in this direction:

- Dominating Set, Independent Dominating Set, Connected Dominating Set, Total Dominating Set, and Acyclic Dominating Set are $W[1]$-hard in circle graphs, parameterized by the size of the solution.
- Whereas both Connected Dominating Set and Acyclic Dominating Set are $W[1]$-hard in circle graphs, it turns out that Connected Acyclic Dominating Set is polynomial-time solvable in circle graphs.
- If T is a *given* tree, deciding whether a circle graph has a dominating set isomorphic to T is NP-complete when T is in the input, and FPT when parameterized by $|V(T)|$. We prove that the FPT algorithm is subexponential.

Keywords: circle graphs, domination problems, parameterized complexity, parameterized algorithms, dynamic programming, constrained domination.

1 Introduction

A *circle graph* is the intersection graph of a set of chords in a circle (see Fig. 1 for an example of a circle graph G together with a circle representation of it). The class of circle graphs has been extensively studied in the literature, due in part to its applications to sorting [12] and VLSI design [27]. Many problems which are NP-hard in general graphs turn out to be solvable in polynomial time

* The third author was partially supported by EPSRC Grant EP/G043434/1. The other authors were partially supported by AGAPE (ANR-09-BLAN-0159) and GRATOS (ANR-09-JCJC-0041) projects (France).

M.C. Golumbic et al. (Eds.): WG 2012, LNCS 7551, pp. 308–319, 2012.

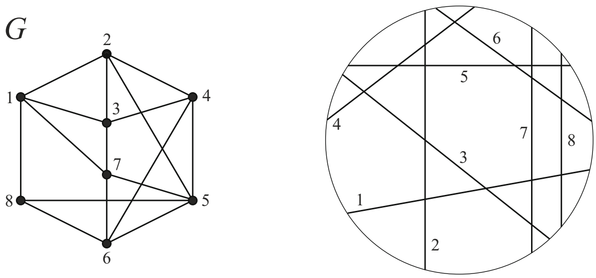

Fig. 1. A circle graph G on 8 vertices together with a circle representation of it

when restricted to circle graphs. For instance, this is the case of Maximum Clique and Maximum Independent Set [17], Treewidth [24], Minimum Feedback Vertex Set [18], Recognition [19,28], Dominating Clique [22], or 3-Colorability [30].

But still a few problems remain NP-complete in circle graphs, like k-Colorability for $k \geq 4$ [29], Hamiltonian Cycle [8], or Minimum Clique Cover [23]. In this article we study a variety of domination problems in circle graphs, from a parameterized complexity perspective. A *dominating set* in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that every vertex in $V \setminus S$ has at least one neighbor in S. Some extra conditions can be imposed to a dominating set. For instance, if $S \subseteq V$ is a dominating set and $G[S]$ is connected (resp. acyclic, an independent set, a graph without isolated vertices, a tree, a path), then S is called a *connected* (resp. *acyclic*, *independent*, *total*, *tree*, *path*) *dominating set*. In the example of Fig. 1, vertices 1 and 5 (resp. 3, 4, and 6) induce an independent (resp. connected) dominating set. The corresponding minimization problems are defined in the natural way. Given a set of graphs $\mathcal{G}$, the Minimum $\mathcal{G}$-Dominating Set problem consists in, given a graph G, finding a dominating set $S \subseteq V(G)$ of G of minimum cardinality such that $G[S]$ is isomorphic to some graph in $\mathcal{G}$. Throughout the article, we may omit the word "Minimum" when referring to a specific problem.

For an introduction to parameterized complexity theory, see for instance [10, 14, 26]. A decision problem with input size n and parameter k having an algorithm which solves it in time $f(k) \cdot n^{\mathcal{O}(1)}$ (for some computable function f depending only on k) is called *fixed-parameter tractable*, or FPT for short. The parameterized problems which are $W[i]$-hard for some $i \geq 1$ are not likely to be FPT [10, 14, 26]. A parameterized problem is in XP if it can be solved in time $f(k) \cdot n^{g(k)}$, for some (unrestricted) functions f and g. The parameterized versions of the above domination problems when parameterized by the cardinality of a solution are also defined naturally.

Previous Work. Dominating Set is one of the most prominent classical graph-theoretic NP-complete problems [16], and has been studied intensively in the literature. Keil [22] proved that Dominating Set, Connected Dominating Set, and Total Dominating Set are NP-complete when restricted to circle graphs, and Damian and Pemmaraju [9] proved that Independent Dominating Set is also NP-complete in circle graphs, answering an open question from Keil [22].

Hedetniemi, Hedetniemi, and Rall [20] introduced acyclic domination in graphs. In particular, they proved that Acyclic Dominating Set can be solved in polynomial time in interval graphs and proper circular-arc graphs. Xu, Kang, and Shan [31] proved that Acyclic Dominating Set is linear-time solvable in bipartite permutation graphs. The complexity status of Acyclic Dominating Set in circle graphs was unknown.

In the theory of parameterized complexity [10,14,26], Dominating Set also plays a fundamental role, being the paradigm of a $W[2]$-hard problem. For some graph classes, like planar graphs, Dominating Set remains NP-complete [16] but becomes FPT when parameterized by the size of the solution [2]. Other more recent examples can be found in H-minor-free graphs [3] and claw-free graphs [7].

The parameterized complexity of domination problems has been also studied in geometric graphs, like k-polygon graphs [11], multiple-interval graphs and their complements [13,21], k-gap interval graphs [15], or graphs defined by the intersection of unit squares, unit disks, or line segments [25]. But to the best of our knowledge, the parameterized complexity of the aforementioned domination problems in circle graphs was open.

Our Contribution. In this paper we prove the following results, which settle the parameterized complexity of a number of domination problems in circle graphs:

- In Section 2, we prove that Dominating Set, Connected Dominating Set, Total Dominating Set, Independent Dominating Set, and Acyclic Dominating Set are $W[1]$-hard in circle graphs, parameterized by the size of the solution. Note that Acyclic Dominating Set was not even known to be NP-hard in circle graphs. The reductions are from k-Colored Clique in general graphs.
- Whereas both Connected Dominating Set and Acyclic Dominating Set are $W[1]$-hard in circle graphs, it turns out that Connected Acyclic Dominating Set is polynomial-time solvable in circle graphs. This is proved in Section 3.
- Furthermore, if T is a *given* tree, we prove in Section 3 that the problem of deciding whether a circle graph has a dominating set isomorphic to T is NP-complete but FPT when parameterized by $|V(T)|$. The NP-completeness reduction is from 3-Partition, and we prove that the running time of the FPT algorithm is subexponential. As a corollary of this algorithm, we also deduce that if T has bounded degree, then deciding whether a circle graph has a dominating set isomorphic to T can be solved in polynomial time.

Due to lack of space, the proofs marked with '[$\star$]' have been omitted in this extended abstract, and can be found in [5].

Further Research. Some interesting questions remain open. We proved that several domination problems are $W[1]$-hard in circle graphs. Are they $W[1]$-complete, or may they also be $W[2]$-hard? On the other hand, we proved that finding a dominating set isomorphic to a tree can be done in polynomial time. It could be interesting to generalize this result to dominating sets isomorphic to a connected graph of fixed treewidth. Finally, even if Dominating Set parameterized by treewidth is FPT in general graphs due to Courcelle's theorem [6], it is not plausible that it has a polynomial kernel in general graphs [4]. It may be the case that the problem admits a polynomial kernel parameterized by treewidth (or by vertex cover) when restricted to circle graphs.

2 $W[1]$-Hardness Results

In this section we prove hardness results for a number of domination problems in circle graphs. In a representation of a circle graph, we will always consider the circle oriented anticlockwise. Given three points a, b, c in the circle, by $a < b < c$ we mean that starting from a and moving anticlockwise along the circle, b comes before c. In a circle representation, we say that two chords with endpoints (a, b) and (c, d) are *parallel twins* if $a < c < d < b$, and there is no other endpoint of a chord between a and c, nor between d and b. Note that for any pair of parallel twins (a, b) and (c, d), we can slide c (resp. d) arbitrarily close to a (resp. b) without modifying the circle representation.

We start with the main result of this section.

Theorem 1. Dominating Set *is $W[1]$-hard in circle graphs, when parameterized by the size of the solution.*

Proof: The reduction is from the k-Colored Clique problem: given a graph $G = (V, E)$ and a coloring of V using k colors, the question is whether there is clique of size k in G containing exactly one vertex from each color. This problem is $W[1]$-hard when parameterized by k [13]. It can be easily seen that we may assume that all color classes are independent sets of the same size. We shall reduce the k-Colored Clique problem to the problem of finding a dominating set of size at most $k(k+1)/2$ in circle graphs. Let k be an integer and let G be a k-colored graph on kn vertices such that n vertices are colored with color i for all $1 \leq i \leq k$. For every $1 \leq i \leq k$, we denote by x^i_j the vertices of color i, with $1 \leq j \leq n$. Let us prove that G has a k-colored clique of size k if and only if the following circle graph C has a dominating set of size at most $k(k+1)/2$. We choose an arbitrary point of the circle as the *origin*. The circle graph C is defined as follows:

- We divide the circle into k disjoint open intervals $]s_i, s'_i[$ for $1 \leq i \leq k$, called *sections*. Each section is divided into $k+1$ disjoint intervals $]c_{ij}, c'_{ij}[$ for $1 \leq j \leq k+1$, called *clusters* (see Fig. 2 for an illustration). Each cluster has n particular points denoted by $1, \ldots, n$ following the order of the circle. These intervals are constructed in such a way that the origin is not in a section.

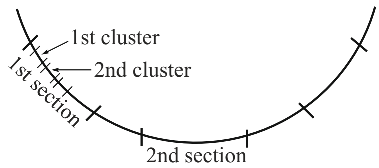

Fig. 2. Sections and clusters in the reduction of Theorem 1

- Sections are numbered from 1 to k following the anticlockwise order from the origin. Similarly, the clusters inside each section are numbered from 1 to $k+1$.
- For each $1 \leq i \leq k, 1 \leq j \leq k+1$, we add a chord with endpoints c_{ij} and c'_{ij}, which we call the *extremal chord* of the j-cluster of the i-th section.
- For each $1 \leq i \leq k$ and $1 \leq j \leq k$, we add chords between the j-th and the $(j+1)$-th clusters of the i-th section as follows. For each $0 \leq l \leq n$, we add two parallel twin chords, each having one endpoint in the interval $]l, l+1[$ of the j-th cluster, and the other endpoint in the interval $]l, l+1[$ of the $(j+1)$-th cluster. These chords are called *inner chords* (see Fig. 3(a) for an illustration). We note that the endpoints of the inner chords inside each interval can be chosen arbitrarily. The interval $]0, 1[$ is the interval between c_{ij} and the point 1, and similarly $]n, n+1[$ is the interval between the point n and c'_{ij}.
- We also add chords between the first and the last clusters of each section. For each $1 \leq i \leq k$ and $1 \leq l \leq n$, we add a chord joining the point l of the first cluster and the point l of the last cluster of the i-th section. For each $1 \leq i \leq k$, these chords are called the *i-th memory chords.*
- Extremal, inner, and memory chords will ensure some structure on the solution. On the other hand, the following chords will simulate the behavior of the original graph. In fact, the n particular points in each cluster of the i-th section will simulate the behavior of the n vertices of color i in G. Let $i < j$. The chords from the i-th section to the j-th section are between the j-th cluster of the i-th section and the $(i+1)$-th cluster of the j-th section. Between this pair of clusters, we add a chord joining the point h (in the i-th section) and the point l (in the j-th section) if and only if $x_h^i x_l^j \in E(G)$. We say that such a chord is called *associated* with an edge of the graph G, and such chords are called *outer chords.* In other words, there is an outer chord in C if the corresponding vertices are connected in G.

Intuitively, the idea of the above construction is as follows. For each $1 \leq i \leq k$, among the $k+1$ clusters in the i-th section, the first and the last one do not contain endpoints of outer chords, and are only used for technical reasons (as discussed below). The remaining $k-1$ clusters in the i-th section capture the edges of G between vertices of color i and vertices of the remaining $k-1$ colors. Namely, for any two distinct colors i and j, there is a cluster in the i-th section and a cluster in the j-th section such that the outer chords between these two

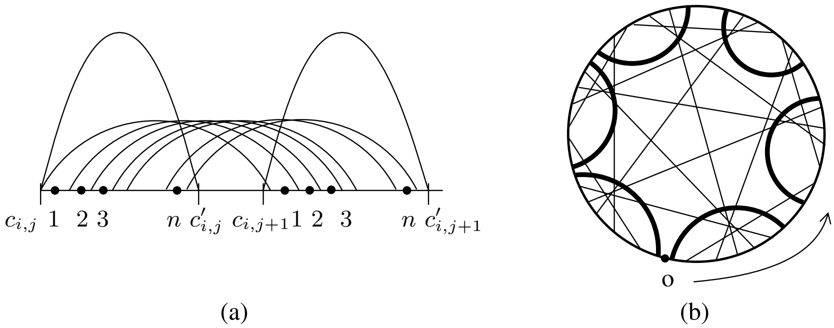

Fig. 3. (a) Representation of the chords between the j-th and the $(j+1)$-th cluster of the i-th section. The higher chords are extremal chords. The others are inner chords and have to be replaced by two parallel twin chords. (b) The general form of a solution. Thick chords are memory chords and the other ones are outer chords. The origin is depicted with a small "o".

clusters correspond to the edges in G between colors i and j. The rest of the proof is structured along a series of claims.

Claim 1. [$\star$] *If there exists a k-colored clique in G, then there exists a dominating set of size $k(k+1)/2$ in C.*

In the following we will state some properties about the dominating sets in C of size $k(k+1)/2$.

Claim 2. [$\star$] *A dominating set in C has size at least $k(k+1)/2$, and a dominating set of this size has exactly one endpoint in each cluster.*

Claim 3. [$\star$] *A dominating set of size $k(k+1)/2$ in C contains no inner nor extremal chord.*

By Claim 3, a dominating set in C of size $k(k+1)/2$ contains only memory and outer chords. Thus, the unique (by Claim 2) endpoint of the dominating set in each cluster is one of the points $\{1, \ldots, n\}$, and we call it the *value* of a cluster. Fig. 3(b) illustrates the general form of a solution.

Claim 4. [$\star$] *Assume that C contains a dominating set of size $k(k+1)/2$. Then, in a given section, the value of a cluster does not increase between consecutive clusters.*

Claim 5. [$\star$] *Assume that C contains a dominating set of size $k(k+1)/2$. Then, for each $1 \leq i \leq k$, all the clusters of the i-th section have the same value.*

The *value* of a section is the value of the clusters in this section (note that it is well-defined by Claim 5). The *vertex associated with the i-th section* is the vertex x^i_k if the value of the i-th section is k.

Claim 6. [$\star$] *If there is a dominating set in C of size $k(k+1)/2$, then for each pair (i,j) with $1 \leq i < j \leq k$, the vertex associated with the i-th section is adjacent in G to the vertex associated with the j-th section. Therefore, G has a k-colored clique.*

Claims 1 and 6 together ensure that C has a dominating set of size $k(k+1)/2$ if and only if G has a k-colored clique. The reduction can be easily done in polynomial time, and the parameters of the problems are polynomially equivalent. Thus, Dominating Set in circle graphs is $W[1]$-hard. This completes the proof of Theorem 1. $\square$

Note that in the construction of Theorem 1, if there is a dominating set of size $k(k+1)/2$ in C, it is necessarily connected (see the form of the solution in Fig. 3(b)). Indeed, the memory chords ensure the connectivity between all the chords with one endpoint in a section. Since there is a chord between each pair of sections, the dominating set is connected. Note also that a connected dominating set is also a total dominating set, as it contains no isolated vertices. Therefore, we obtain the following corollary.

Corollary 1. Connected Dominating Set *and* Total Dominating Set *are $W[1]$-hard in circle graphs, when parameterized by the size of the solution.*

In the following hardness result, we use a completely different reduction from k-Colored Clique.

Theorem 2. [$\star$] Independent Dominating Set *is $W[1]$-hard in circle graphs.*

The construction of Theorem 2 can be appropriately modified to deal with the case when the dominating set is required to induce an acyclic subgraph.

Theorem 3. [$\star$] Acyclic Dominating Set *is $W[1]$-hard in circle graphs.*

3 Tree Dominating Sets

In this section we focus on finding dominating sets in a circle graph which induce graphs isomorphic to trees. Namely, in Theorem 4 we give a polynomial-time algorithm to find a dominating set isomorphic to *some* tree. We prove in Theorem 5 that finding a dominating set isomorphic to a *given* tree is NP-complete. In Theorem 6 we modify the algorithm of Theorem 4 to find a dominating set isomorphic to a *given* tree T in FPT time, the parameter being the size of T. By carefully analyzing its running time, we prove that this FPT algorithm runs in *subexponential* time. It also follows from this analysis that if the given tree T has bounded degree (in particular, if it is a path), then the problem of find a dominating set isomorphic to T can be solved in polynomial time. Note that, in contrast with Theorem 4 below, Theorem 3 in Section 2 states that, if $\mathcal{F}$ is the set of all forests, then $\mathcal{F}$-Dominating Set is $W[1]$-hard in circle graphs.

Theorem 4. *Let $\mathcal{T}$ be the set of all trees. Then* $\mathcal{T}$-Dominating Set *can be solved in polynomial time in circle graphs. In other words,* Connected Acyclic Dominating Set *can be solved in polynomial time in circle graphs.*

Proof: Let C be a circle graph on n vertices, and let $\mathcal{C}$ be a circle representation of C. We denote by $\mathcal{P}$ the set of intersections of the circle and the chords in this representation. The elements of $\mathcal{P}$ are called *points*. W.l.o.g., we can assume that only one chord intersects a given point. Given two points $a, b \in \mathcal{P}$, the *interval* $[a, b]$ is the interval from a to b in the anticlockwise order. Given four (non-necessarily distinct) points $a, b, c, d \in \mathcal{P}$, with $a \le c \le d \le b$, by the *region* $ab - cd$ we mean the union of the two intervals $[a, c]$ and $[d, b]$. Note that these two intervals can be obtained by "subtracting" the interval $[c, d]$ from the interval $[a, b]$; this is why we use the notation $ab - cd$.

In the following, by *size* of a set of chords we mean the number of vertices of C in this set. We say that a forest F of C *spans* a region $ab - cd$ if each of a, b, c, and d is an endpoint of some chord in F, and each endpoint of a chord of F is either in $[a, c]$ or in $[d, b]$. A forest F is *split* by a region $ab - cd$ if for each connected component of F there is exactly one chord with one endpoint in $[a, c]$ and one endpoint in $[d, b]$. Given a region $ab - cd$, a forest F is $(ab - cd)$-*dominating* if all the chords of C with both endpoints either in the interval $[a, c]$ or in the interval $[d, b]$ are dominated by F. A forest is *valid* for a region $ab - cd$ if it spans $ab - cd$, is split by $ab - cd$, and is $(ab - cd)$-dominating.

Note that an $(ab - cd)$-dominating forest with several connected components might not dominate some chord going from $[a, c]$ to $[d, b]$. This is not the case if F is connected, as stated in the following claim.

Claim 7. [$\star$] *Let T be a valid tree for a region $ab - cd$. Then all the chords of C with both endpoints in $[a, c] \cup [d, b]$ are dominated by T.*

We now state two properties that will be useful in the algorithm. Their correctness is proved below.

- **T1.** Let F_1 and F_2 be two valid forests for two regions $ab - cd$ and $ef - gh$, respectively, such that $a \le c \le e \le g \le h \le f \le d \le b$. If there is no chord with both endpoints either in $[c, e]$ or in $[f, d]$, then $F_1 \cup F_2$ is valid for $ab - gh$ (see Fig. 4).
- **T2.** Let F_1 and F_2 be two valid forests for two regions $ab - cd$ and $ef - gh$, respectively (F_2 being possibly empty), and let uv be a chord such that $u \le a \le c \le e \le g \le v \le h \le f \le d \le b$, and such that there is no chord with both endpoints either in $[u, a]$, or in $[g, v]$, or in $[v, h]$, or in $[b, u]$. Then $F_1 \cup F_2 \cup \{uv\}$ is a tree which is valid for $df - ce$. When F_2 is empty, we consider that e, f, g, h correspond to the point v. (see Fig. 4).

Roughly speaking, the intuitive idea behind this two properties is to reduce the length of the circle in which we still have to do some computation (that is, outside the valid regions). Again, the proof is structured along a series of claims. Before verifying the correctness of Properties **T1** and **T2**, let us first state a useful general fact.

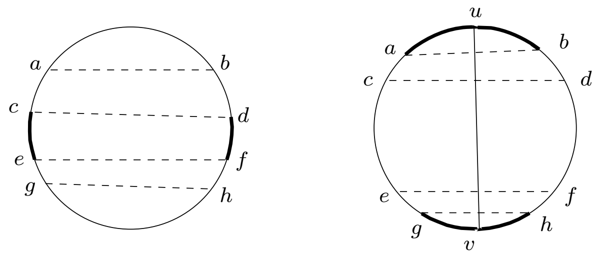

Fig. 4. On the left (resp. right), regions corresponding to Property **T1** (resp. Property **T2**). Full lines correspond to real chords of C, dashed lines correspond to the limit of regions. Bold intervals correspond to intervals with no chord of C with both endpoints in the interval.

Claim 8. [$\star$] *Let $ab-cd$ be a region and let F be a valid forest for $ab-cd$. The chords with one endpoint in $[c,d]$ and one endpoint in $[d,c]$ are dominated by F.*

Claim 9. [$\star$] *Properties* **T1** *and* **T2** *are correct.*

For a region $ab-cd$, we denote by $v^f_{ab,cd}$ (resp. $v^t_{ab,cd}$) the least integer l for which there is a valid forest (resp. tree) of size l for $ab-cd$. If there is no valid forest (resp. tree) for $ab-cd$, we set $v^f_{ab,cd} = +\infty$ (resp. $v^t_{ab,cd} = +\infty$). Let us now describe our algorithm based on dynamic programming. With each region $ab-cd$, we associate two integers $v^1_{ab,cd}$ and $v^2_{ab,cd}$. Algorithm 1 below calculates these two values for each region. We next show that $v^1_{ab,cd} = v^f_{ab,cd}$ and $v^2_{ab,cd} = v^t_{ab,cd}$, and that Algorithm 1 correctly computes the result in polynomial time.

Algorithm 1. Dynamic programming for computing a dominating tree

for each region $ab-cd$ **do** $v^1_{ab,cd} \leftarrow \infty$; $v^2_{ab,cd} \leftarrow \infty$
for each chord ab of the circle graph **do** $v^1_{ab,ab} \leftarrow 1$; $v^2_{ab,ab} \leftarrow 1$
for $j = 2$ to n **do**
  if there are two regions $ab-cd$ and $ef-gh$ such that $v^1_{ab,cd} = j_1$ and $v^1_{ef,gh} = j_2$ with $j_1 + j_2 = j$ satisfying Property **T1**, with $v^1_{ab,gh} = +\infty$ **then**
    $v^1_{ab,gh} \leftarrow j$
  if there is a region $ab-cd$ and a chord uv such that $v^1_{ab,cd} = j-1$ satisfying Property **T2** with an empty second forest **then**
    if $v^1_{dv,cv} = +\infty$ **then**
      $v^1_{dv,cv} \leftarrow j$
    if $v^2_{dv,cv} = +\infty$ **then**
      $v^2_{dv,cv} \leftarrow j$
  if there are two regions $ab-cd$ and $ef-gh$ and a chord uv such that $v^1_{ab,cd} = j_1$ and $v^1_{ef,gh} = j_2$ with $j_1 + j_2 = j-1$ satisfying Property **T2 then**
    if $v^1_{df,ce} = +\infty$ **then**
      $v^1_{df,ce} \leftarrow j$
    if $v^2_{df,ce} = +\infty$ **then**
      $v^2_{df,ce} \leftarrow j$

Claim 10. [$\star$] *For any region $ab - cd$, $v^1_{ab,cd} = v^f_{ab,cd}$ and $v^2_{ab,cd} = v^t_{ab,cd}$.*

Let us now explain how we can verify if there is a dominating set in C isomorphic to some tree of a given size.

Claim 11. [$\star$] *Let k be a positive integer. There is a dominating tree of size at most k in C if and only if there is a region $ab - cd$ such that $v^t_{ab,cd} \leq k$ and such that there is no chord strictly contained in $[b, a]$ nor in $[c, d]$.*

By Claims 10 and 11, it follows that Algorithm 1 computes the regions for which there is a valid tree of any size from 1 to n. Given a region $ab - cd$ with $v^t_{ab,cd} \leq k$, we just have to verify that there are no chords in the intervals $[b, a]$ and $[c, d]$, which can clearly be done in polynomial time. One can easily check that Algorithm 1 runs in time $\mathcal{O}(n^{10})$, but we did not make any attempt to improve its time complexity. $\square$

It turns out that when we seek a dominating set isomorphic to a *given* tree T, the problem is NP-complete. The reduction is from the 3-Partition problem, which consists in deciding whether a given multiset of integers can be partitioned into triples such that the three integers in each triple have the same sum.

Theorem 5. [$\star$] *Let T be a given tree. Then* $\{T\}$-Dominating Set *is* NP-*complete in circle graphs when T is part of the input.*

Finally, we show that $\{T\}$-Dominating Set in circle graphs can be solved by a subexponential FPT algorithm, when parameterized by $|V(T)|$.

The algorithm of Theorem 6 below goes along the same lines of Algorithm 1 given in the proof of Theorem 4. The main difference is that in the proof of Theorem 4, when Properties **T1** or **T2** are satisfied, we can directly apply them and still obtain a forest or a tree. But when looking for a given tree T, when we make the union of two forests, we have to make sure that the union of these two forests is still a subforest of T, and that we can correctly complete it to obtain the desired tree T. For obtaining that, we will apply two new properties corresponding to Properties **T1** or **T2**, whenever it is possible to create forests which are induced by the children of the same vertex of T.

Let us give some more intuition on the algorithm. We consider the tree T rooted at an arbitrary vertex r. Let v be a vertex of T, and let $w_1, \ldots, w_l$ be the children of v. We define $T(v)$ as the forest $T[w_1] \cup T[w_2] \ldots \cup T[w_l]$, where $T[w_i]$ is the subtree of the rooted tree T induced by w_i and the descendants of w_i. Roughly speaking, the idea of the algorithm is to exhaustively seek a dominating set isomorphic to any possible subforest of $F(v)$ for every vertex v in T, and then try to grow it until hopefully obtaining the target tree T. Note that if a vertex v of T has k children, there are a priori 2^k possible subsets of children of y, which define 2^k possible types of subforests in $F(v)$. But the key point in order to obtain a subexponential algorithm is that if some of the trees in $F(v)$ are isomorphic, some of the choices of subsets of subforests will give rise to the same tree. In order to avoid this redundancy, for each vertex v of T, we partition the trees in $F(v)$ into isomorphism classes, and then the choices within each isomorphism class reduce to choosing the multiplicity of this tree. Note that carrying out this partition into isomorphism classes can be done in polynomial time (in the size of T) for each vertex of T, using the fact that one can test whether two rooted trees T_1 and T_2 with t vertices are isomorphic in $\mathcal{O}(t)$ time [1]. The details can be found in [5].

From our analysis, it also follows that if T has bounded degree (in particular, if it is a path), then $\{T\}$-Dominating Set can be solved in polynomial time in circle graphs.

Theorem 6. [$\star$] *Let T be a given tree. There exists an* FPT *algorithm to solve* $\{T\}$-Dominating Set *in a circle graph on n vertices, when parameterized by $t = |V(T)|$, running in time $2^{\mathcal{O}(t \cdot \frac{\log \log t}{\log t})} \cdot n^{\mathcal{O}(1)} = 2^{o(t)} \cdot n^{\mathcal{O}(1)}$. Furthermore, if T has bounded degree, then* $\{T\}$-Dominating Set *can be solved in polynomial time in circle graphs.*

Acknowledgment. We would like to thank Sylvain Guillemot for stimulating discussions that motivated some of the research carried out in this paper.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley (1974)
2. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed Parameter Algorithms for Dominated Set and Related Problems on Planar Graphs. Algorithmica 33(4), 461–493 (2002)
3. Alon, N., Gutner, S.: Kernels for the Dominating Set Problem on Graphs with an Excluded Minor. Electronic Colloquium on Computational Complexity (ECCC) 15(066) (2008)
4. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. Journal of Computer and System Sciences 75(8), 423–434 (2009)
5. Bousquet, N., Gonçalves, D., Mertzios, G.B., Paul, C., Sau, I., Thomassé, S.: Parameterized Domination in Circle Graphs. Manuscript available at `http://arxiv.org/abs/1205.3728` (2012)
6. Courcelle, B.: The Monadic Second-Order Logic of Graphs: Definable Sets of Finite Graphs. In: van Leeuwen, J. (ed.) WG 1988. LNCS, vol. 344, pp. 30–53. Springer, Heidelberg (1989)
7. Cygan, M., Philip, G., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: Dominating set is fixed parameter tractable in claw-free graphs. Theoretical Computer Science 412(50), 6982–7000 (2011)
8. Damaschke, P.: The Hamiltonian Circuit Problem for Circle Graphs is NP-Complete. Information Processing Letters 32(1), 1–2 (1989)
9. Damian-Iordache, M., Pemmaraju, S.V.: Hardness of Approximating Independent Domination in Circle Graphs. In: Aggarwal, A.K., Pandu Rangan, C. (eds.) ISAAC 1999. LNCS, vol. 1741, pp. 56–69. Springer, Heidelberg (1999)
10. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, New York (1999)

11. Elmallah, E.S., Stewart, L.K.: Independence and domination in polygon graphs. Discrete Applied Mathematics 44(1-3), 65–77 (1993)
12. Even, S., Itai, A.: Queues, stacks and graphs. In: Press, A. (ed.) Theory of Machines and Computations, pp. 71–86 (1971)
13. Fellows, M.R., Hermelin, D., Rosamond, F.A., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. Theoretical Computer Science 410(1), 53–61 (2009)
14. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer (2006)
15. Fomin, F., Gaspers, S., Golovach, P., Suchan, K., Szeider, S., Jan Van Leeuwen, E., Vatshelle, M., Villanger, Y.: k-Gap Interval Graphs. In: Fernández-Baca, D. (ed.) LATIN 2012. LNCS, vol. 7256, pp. 350–361. Springer, Heidelberg (2012), `http://arxiv.org/abs/1112.3244`
16. Garey, M., Johnson, D.: Computers and Intractability. W.H. Freeman, San Francisco (1979)
17. Gavril, F.: Algorithms for a maximum clique and a maximum independent set of a circle graph. Networks 3, 261–273 (1973)
18. Gavril, F.: Minimum weight feedback vertex sets in circle graphs. Information Processing Letters 107(1), 1–6 (2008)
19. Gioan, E., Paul, C., Tedder, M., Corneil, D.: Circle Graph Recognition in Time $O(n+m) \cdot \alpha(n+m)$. Manuscript available at `http://arxiv.org/abs/1104.3284` (2011)
20. Hedetniemi, S.M., Hedetniemi, S.T., Rall, D.F.: Acyclic domination. Discrete Mathematics 222(1-3), 151–165 (2000)
21. Jiang, M., Zhang, Y.: Parameterized Complexity in Multiple-Interval Graphs: Domination. In: Marx, D., Rossmanith, P. (eds.) IPEC 2011. LNCS, vol. 7112, pp. 27–40. Springer, Heidelberg (2012)
22. Keil, J.M.: The complexity of domination problems in circle graphs. Discrete Applied Mathematics 42(1), 51–63 (1993)
23. Keil, J.M., Stewart, L.: Approximating the minimum clique cover and other hard problems in subtree filament graphs. Discrete Applied Mathematics 154(14), 1983–1995 (2006)
24. Kloks, T.: Treewidth of circle graphs. International Journal of Foundations of Computer Science 7(2), 111–120 (1996)
25. Marx, D.: Parameterized Complexity of Independence and Domination on Geometric Graphs. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 154–165. Springer, Heidelberg (2006)
26. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006)
27. Sherwani, N.A.: Algorithms for VLSI Physical Design Automation. Kluwer Academic Press (1992)
28. Spinrad, J.: Recognition of circle graphs. Journal of Algorithms 16(2), 264–282 (1994)
29. Unger, W.: On the k-Colouring of Circle-Graphs. In: Cori, R., Wirsing, M. (eds.) STACS 1988. LNCS, vol. 294, pp. 61–72. Springer, Heidelberg (1988)
30. Unger, W.: The Complexity of Colouring Circle Graphs (Extended Abstract). In: Finkel, A., Jantzen, M. (eds.) STACS 1992. LNCS, vol. 577, pp. 389–400. Springer, Heidelberg (1992)
31. Xu, G., Kang, L., Shan, E.: Acyclic domination on bipartite permutation graphs. Information Processing Letters 99(4), 139–144 (2006)