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, United Kingdom.
`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 [13] and VLSI design [31]. Many problems which are NP-hard in general graphs turn out to be solvable in polynomial time when restricted to circle graphs. For instance, this is the case of MAXIMUM CLIQUE and MAXIMUM INDEPENDENT SET [19], TREEWIDTH [26], MINIMUM FEEDBACK VERTEX SET [20], RECOGNITION [21,32], DOMINATING CLIQUE [24], or 3-COLORABILITY [34].

But still a few problems remain NP-complete in circle graphs, like k-COLORABILITY for $k \geq 4$ [33], HAMILTONIAN CYCLE [8], or MINIMUM CLIQUE COVER [25]. 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,

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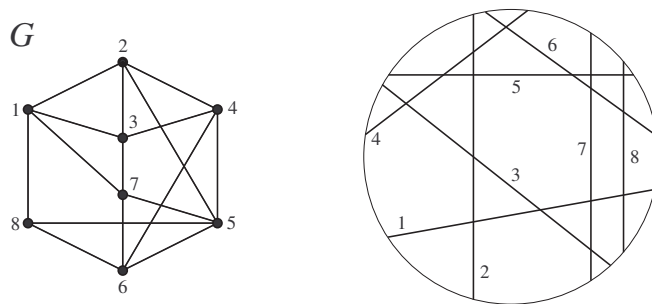
Fig. 1. A circle graph G on 8 vertices together with a circle representation of it.

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, 16, 28]. 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, 16, 28]. 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 [18], and has been studied intensively in the literature. Keil [24] 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 [24].

Hedetniemi, Hedetniemi, and Rall [22] 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 [35] 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, 16, 28], 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 [18] 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 [14, 23], k-gap interval graphs [17], or graphs defined by the intersection of unit squares, unit disks, or line segments [27]. 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 3.1.
- Furthermore, if T is a *given* tree, we prove that the problem of deciding whether a circle graph has a dominating set isomorphic to T is NP-complete (Section 2.3) but FPT when parameterized by $|V(T)|$ (Section 3.2). 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 the algorithm presented in Section 3.2, 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.

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 [5]. 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 Hardness results

In this section we prove hardness results for a number of domination problems in circle graphs. In order to prove the $W[1]$-hardness of the domination problems, we provide two families of reductions. Namely, in Section 2.1 we prove the hardness of DOMINATING SET, CONNECTED DOMINATING SET, and TOTAL DOMINATING SET, and in Section 2.2 we prove the hardness of INDEPENDENT DOMINATING SET and ACYCLIC DOMINATING SET. Finally, we prove the NP-completeness for trees in Section 2.3.

For better visibility, some figures of this section have colors, but these colors are not indispensable for completely understanding the depicted constructions. Before stating the hardness results, we need to introduce the following parameterized problem, proved to be $W[1]$-hard in [14].

k-COLORED CLIQUE
    Instance: A graph $G = (V, E)$ and a coloring of V using k colors.
Parameter: k.
    Question: Does there exist a clique of size k in G containing
                exactly one vertex from each color?

Note that in an instance of k-COLORED CLIQUE, we can assume that there is no edge between any pair of vertices colored with the same color. Also, we can assume that for each $1 \le i \le k$, the number of vertices colored with color i is the same. Indeed, given an
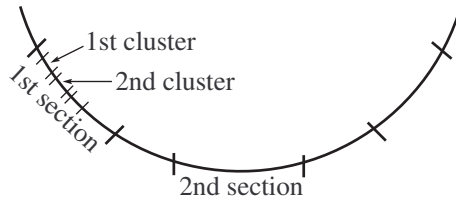
Fig. 2. Sections and clusters in the reduction of Theorem 1.

instance G, we can consider an equivalent instance G' obtained by putting together $k!$ disjoint copies of G, one for each permutation of the color classes.

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.

### 2.1   Hardness of domination, and connected and total domination

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: 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_j^i 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.
- 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 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.
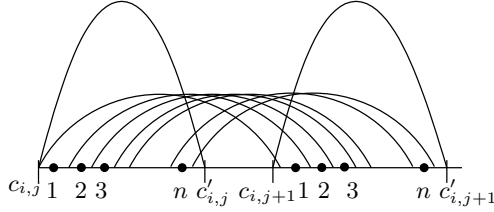

Fig. 3. Representation of the chords between the j-th and the $(j+1)$-th cluster of the i-th section. The higher chords are extremal chords. The others are inner chords and have to be replaced by two parallel twin chords.

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

Proof: Assume that there is a k-colored clique K in G and let us denote by k_i the integer such that $x_{k_i}^i$ is the vertex of color i in this clique. Let $\mathcal{D}$ be the following set of chords. For each section $1 \leq i \leq k$, we add to $\mathcal{D}$ the memory chord joining the points k_i of the first and the last clusters. We also add in $\mathcal{D}$ the outer chords associated with the edges of the k-colored clique. The set $\mathcal{D}$ contains $k(k+1)/2$ chords: k memory chords and $k(k-1)/2$ outer chords. Let us prove that $\mathcal{D}$ is a dominating set.

The extremal chords are dominated, since $\mathcal{D}$ has exactly one endpoint in each cluster. Indeed, there is an endpoint in the first and the last cluster of section because of the memory chords of $\mathcal{D}$. There is an endpoint in the other clusters because of the outer chord associated with the edge of the k-colored clique. The inner chords are also dominated. Indeed, for each section i, the endpoint of the chord of $\mathcal{D}$ is k_i, for all clusters j such that $1 \leq j \leq k+1$. Thus, for all $1 \leq l \leq n$, the inner chords between the intervals $]l, l+1[$ of
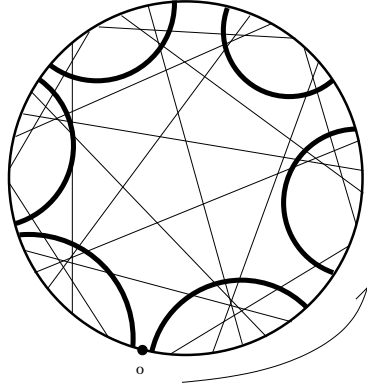
Fig. 4. The general form of a solution in the reduction of Theorem 1. The thick chords are memory chords and the other ones are outer chords. The origin is depicted with a small "o".

the j-th cluster and $]l, l + 1[$ of the $(j + 1)$-th cluster are dominated by the chord of $\mathcal{D}$ with endpoint in the j-th section if $l \leq k_i - 1$, or by the chord of the $(j + 1)$-th section otherwise.

The outer chords are dominated by the memory chords of $\mathcal{D}$, since the outer chords have their endpoints in two different sections. Finally, the memory chords are also dominated by the outer chords of $\mathcal{D}$ for the same reason.   □

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

Claim 2 *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.*

Proof: The *interval of a chord* linking x and y, with $x < y$, is the interval $[x, y]$. One can note that when ℓ chords have pairwise disjoint intervals, at least $\lceil \ell/2 \rceil$ chords are necessary to dominate them.

The intervals $[c_{i,j}, c'_{i,j}]$ of the extremal chords are disjoint by assumption. Since there are $k(k + 1)$ extremal chords, a dominating set has size at least $k(k + 1)/2$. And if there is a dominating set of such size, it must have exactly one endpoint in each interval, i.e., one endpoint in each cluster.   □

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

Proof: Let $\mathcal{D}$ be a dominating set in C of size $k(k + 1)/2$. It contains no extremal chords, since both endpoints of an extremal chord are in the same cluster, which is impossible by Claim 2. If $\mathcal{D}$ contains an inner chord c, the parallel twin of c in C is dominated by some other chord c'. But then $c \cup c'$ intersect at most three clusters, which is again impossible by Claim 2.   □

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. 4 illustrates the general form of a solution.

Claim 4 *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.*

Proof: Assume that in a given arbitrary section, the value of the j-th cluster is l. The inner chords between the interval $]l, l + 1[$ of the j-th cluster and the interval $]l, l + 1[$ of the $(j + 1)$-th cluster have to be dominated. Since the value of the j-th cluster is l, they are not dominated in the j-th cluster. Therefore, in order to ensure the domination of these chords, the value of the $(j+1)$-th cluster is at most l. $\square$

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

Proof: Let $\mathcal{D}$ be such a dominating set. In a given section, the endpoints of $\mathcal{D}$ in the first and the last clusters are endpoints of a memory chord, and for all l, they link the point l of the first cluster to the point l of the last one. Thus, the first and the last clusters have the same value. Since by Claim 4 the value of a cluster decreases between consecutive clusters, the value of the clusters of the same section is necessarily constant. $\square$

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_k^i if the value of the i-th section is k.

Claim 6 *If there is a dominating set in C of size $k(k + 1)/2$, then for each pair (i, j) with $1 \le i < j \le k$, 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.*

Proof: Let i and j be two sections with $i < j$, and let x_k^i and x_l^j be the vertices associated with these two sections, respectively. By Claim 5, the chord of the dominating set in the j-th cluster of the i-th section has a well-defined endpoint k, and the chord of the dominating set in the $(i - 1)$-th cluster of the j-th section has a well-defined endpoint l. The vertex x_k^i associated with the i-th section is adjacent in G to the vertex x_l^j associated with the j-th section. Indeed, the chords having endpoints in these clusters are exactly the chords between these two clusters, and there is a chord if and only if there is an edge between the corresponding vertices in G. $\square$

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. $\square$

From Theorem 1 we can easily deduce the $W[1]$-hardness of two other domination problems in circle graphs.

Corollary 1. CONNECTED DOMINATING SET *and* TOTAL DOMINATING SET *are $W[1]$-hard in circle graphs, when parameterized by the size of the solution.*

Proof: 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. 4). 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. Finally, note that a connected dominating set is also a total dominating set, as it contains no isolated vertices. $\square$
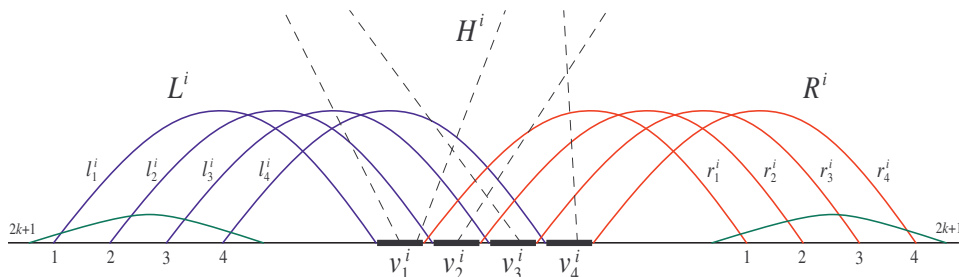
Fig. 5. Gadget H^i in interval I^i used in the proof of Theorem 2, corresponding to a color class X^i of the k-colored input graph G. The dashed chords correspond to non-edges of G.

2.2   Hardness of independent and acyclic domination

We proceed to describe our second construction in order to prove parameterized reductions for domination problems in circle graphs.

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

Proof: We present a parameterized reduction from k-Colored Clique in a general graph to the problem of finding an independent dominating set of size at most $2k$ in a circle graph. Let G be the input k-colored graph with color classes $X^1, \dots, X^k \subseteq V(G)$. Let $x_1^i, \dots, x_n^i$ be the vertices belonging to the color class $X^i \subseteq V(G)$, in an arbitrary order. We proceed to build a circle graph H by defining its circle representation. Let $I^1, \dots, I^k$ be a collection of k disjoint intervals in the circle, which will we associated with the k colors. For $1 \leq i \leq k$, we proceed to construct an induced subgraph H^i of H whose chords have all endpoints in the interval I^i, which we visit from left to right. Throughout the construction, cf. Fig. 5 for an example with $n = 4$.

We start by adding two cliques on n vertices L^i and R^i, with chords $l_1^i, \dots, l_n^i$ and $r_1^i, \dots, r_n^i$, respectively, in the following way. The endpoints of L^i and R^i are placed in three disjoint subintervals of I^i, such that the first subinterval contains, in this order, the left endpoints of $l_1^i, \dots, l_n^i$. The second subinterval contains the right endpoints of L^i and the left endpoints of R^i, in the order $l_1^i, r_1^i, l_2^i, r_2^i, \dots, r_{n-1}^i, l_n^i, r_n^i$. Finally, the third subinterval contains, in this order, the right endpoints of $r_1^i, \dots, r_n^i$. The blue (resp. red) chords in Fig. 5 correspond to L^i (resp. R^i).

For $1 \leq j \leq n$, we define the interval v_j^i as the open interval between the right endpoint of l_j^i and the left endpoint of r_j^i; cf. the thick intervals in Fig. 5. Such an interval v_j^i will correspond to vertex x_j^i of G.

We also add two sets of $2k + 1$ parallel twin chords whose left endpoints are placed exactly before the left (resp. right) endpoint of l_1^i (resp. r_1^i) and whose right endpoints are placed exactly after the left (resp. right) endpoint of l_n^i (resp. r_n^i); cf. the green chords in Fig. 5. We call these chords *parallel chords*. This completes the construction of H^i.

Finally, for each pair of vertices x_p^i, x_q^j of G such that $i \neq j$ and $\{x_p^i, x_q^j\} \notin E(G)$, we add to H a chord $c_{p,q}^{i,j}$ between the interval v_p^i in H^i and the interval v_q^j in H^j; cf. the dashed chords in Fig. 5. We call these chords *outer chords*. That is, the outer chords of H correspond to non-edges of G. This completes the construction of the circle graph H.

We now claim that G has a k-colored clique if and only if H has an independent dominating set of size at most $2k$.

Indeed, let first K be a k-colored clique in G containing vertices $x_{j_1}^1, x_{j_2}^2, \dots, x_{j_k}^k$, and let us obtain from K an independent dominating set S in H. For $1 \leq i \leq k$, the set S contains the two chords $l_{j_i}^i$ and $r_{j_i}^i$ from H^i. Note that S is indeed an independent
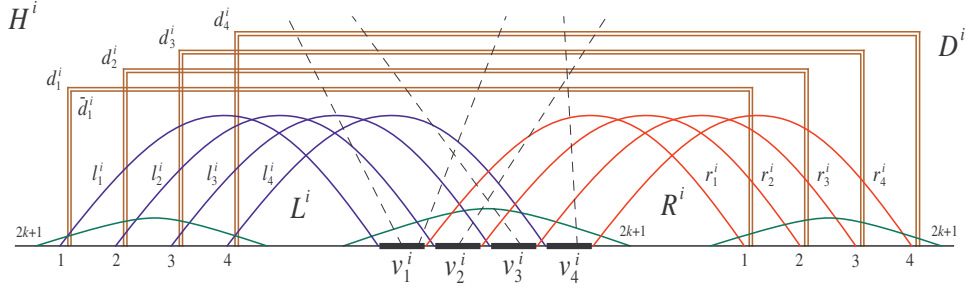
Fig. 6. Gadget H^i in interval I^i used in the proof of Theorem 3, corresponding to a color class X^i of the k-colored input graph G. The dashed chords correspond to non-edges of G.

set. For $1 \leq i \leq k$, since both L^i and R^i are cliques, all the chords in L^i and R^i are dominated by S. Clearly, all parallel chords are also dominated by S. The only outer chords with one endpoint in H^i which are neither dominated by $l^i_{j_i}$ nor by $r^i_{j_i}$ are those with its endpoint in the interval $v^i_{j_i}$. Let c be such an outer chord, and suppose that the other endpoint of c is in H^ℓ. As K is a clique in G, it follows that there is no outer chord in H with one endpoint in $v^i_{j_i}$ and the other in $v^\ell_{j_\ell}$, and therefore necessarily the chord c is dominated either by $l^i_{j_i}$ or by $r^\ell_{j_\ell}$.

Conversely, assume that H has an independent dominating set S with $|S| \leq 2k$. Note that for $1 \leq i \leq k$, because of the two sets of $2k + 1$ parallel chords in H^i, at least one of the chords in L^i and at least one of the chords in R^i must belong to S, so $|S| \geq 2k$. Therefore, it follows that $|S| = 2k$ and that S contains in H^i, for $1 \leq i \leq k$, a pair of non-crossing chords in L^i and R^i. Note that in each H^i, the two chords belonging to S must leave uncovered at least one of the intervals (corresponding to vertices) $v^i_1, \ldots, v^i_n$. Let $v^i_{j_i}$ and $v^\ell_{j_\ell}$ be two uncovered vertices in two distinct intervals I^i and I^ℓ, respectively. By the construction of H, it holds that the vertices $x^i_{j_i}$ and $x^\ell_{j_\ell}$ must be adjacent in G, as otherwise the outer chord in H between the intervals $v^i_{j_i}$ and $v^\ell_{j_\ell}$ would not be dominated by S. Hence, a k-colored clique in G can be obtained by selecting in each H^i any of the uncovered vertices. $\square$

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. ACYCLIC DOMINATING SET *is $W[1]$-hard in circle graphs.*

Proof: As in Theorem 2, the reduction is again from k-COLORED CLIQUE. From a k-colored G, we build a circle graph H that contains all the chords defined in the proof of Theorem 2, plus the following ones for each H^i, $1 \leq i \leq k$ (cf. Fig. 6 for an illustration): we add another set of $2k + 1$ parallel chords whose left (resp. right) endpoints are placed exactly before (resp. after) the right (resp. left) endpoint of l^i_1 (resp. r^i_n); cf. the middle green chords in Fig. 6. We call these three sets of $2k + 1$ chords *parallel* chords.

Furthermore, we add a clique with n chords $d^i_1, \ldots, d^i_n$ such that for $1 \leq j \leq n$ the left (resp. right) endpoint of d^i_j is placed exactly after the left (resp. right) endpoint of l^i_j (resp. r^i_j). Finally, for each such a chord d^i_j we add a parallel twin chord, denoted by $\tilde{d}^i_j$. We call these $2t$ chords *distance* chords, and their union is denoted by D^i; cf. the brown edges in Fig. 6. This completes the construction of H. Note that a pair of chords $l^i_{j_1}$ and $r^i_{j_2}$ dominates all the distance chords in H^i if and only if $l^i_{j_1}$ and $r^i_{j_2}$ do not cross, that is, if and only if $j_1 \leq j_2$.

We now claim that G has a k-colored clique if and only if H has an acyclic dominating set of size at most $2k$.

Indeed, let first K be a k-colored clique in G. An independent (hence, acyclic) dominating set S in H of size $2k$ can be obtained from K exactly as explained in the proof of Theorem 2. Note that in each H^i, the distance chords in D^i are indeed dominated by S because the corresponding chords in L^i and R^i do not cross.

Conversely, assume that H has an acyclic dominating set S with $|S| \leq 2k$. First assume that S contains no outer chord. By the parallel chords in each H^i (cf. the green chords in Fig. 6), it is easy to check that S must contain at least two chords in each H^i, and therefore we have that $|S| = 2k$. We now distinguish several cases according to which two chords in a generic H^i can belong to S. Let $\{u, v\} = S \cap V(H^i)$. Because of the parallel chords, it is clear that only one L^i, R^i, or D^i cannot contain both u and v. It is also clear that no parallel chord can be in S. If $u \in D^i$ and $v \in L^i$, let w.l.o.g. $u = d^i_{j_1}$ and $v = l^i_{j_2}$. Since v must dominate the twin chord $\bar{d}^i_{j_1}$, it follows by the construction of H^i that $j_2 \leq j_1$, and therefore the chord $r^i_{j_1}$ is dominated neither by u nor by v (cf. Fig. 6), a contradiction. The case $u \in D^i$ and $v \in R^i$ is similar. Therefore, we may assume w.l.o.g. that $u = l^i_{j_1}$ and $u = r^i_{j_2}$. Note that such a pair of chords $l^i_{j_1}$ and $r^i_{j_2}$ dominates all the distance chords in H^i if and only if $l^i_{j_1}$ and $r^i_{j_2}$ do not cross. Hence, as in the proof of Theorem 2, for each H^i, $1 \leq i \leq k$, the two chords belonging to S leave at least one uncovered interval $v^i_{j_i}$ (corresponding to vertex $x^i_{j_i}$), and in order for all the outer chords in H to be dominated, the union of the k uncovered vertices must induce a k-colored clique in G. Therefore, if H has an acyclic dominating set of size at most $2k$ with no outer chord, then G has a k-colored clique. Note that in this case S consists of an independent set.

Otherwise, the acyclic dominating set S contains some outer chord. Assume w.l.o.g. that S contains outer chords with at least one endpoint in each of $H^1, \ldots, H^p$ (with $p \geq 2$, as each chord has two endpoints), and no outer chord with an endpoint in any of $H^{p+1}, \ldots, H^k$ (only if $p < k$). By the arguments above, for $p+1 \leq i \leq k$ it follows that S contains exactly one chord in L^i and exactly one chord in R^i. For $1 \leq i \leq p$, in order for all the parallel chords in H^i to be covered, S must contain some chords in L^i, R^i, or D^i. As by assumption $|S| \leq 2k$, in at least one H^i with $1 \leq i \leq p$, S contains exactly one chord in L^i, R^i, or D^i. By the construction of H, this chord must necessarily be a distance chord, as otherwise some parallel chords in H^i would not be dominated by S (cf. Fig. 6). Assume w.l.o.g. that in H^1 only the distance chord $d^1_{j_1}$ and one outer chord outgoing from interval $v^1_{j_2}$ belong to S. But then if $j_2 \geq j_1$ (resp. $j_2 < j_1$) the chord $r^1_{j_2}$ (resp. $l^1_{j_2}$) is not dominated by S, a contradiction. We conclude that S contains at least two outer chords in H^1.

By a simple counting argument, as $|S| \leq 2k$ it follows that for $1 \leq i \leq p$, S contains exactly one distance chord and two outer chords from each of $H^1, \ldots, H^p$ (and, in particular, $|S| = 2k$). But then the subgraph of $H[S]$ induced by the chords belonging to $V(H^1) \cup \ldots \cup V(H^p)$ has minimum degree at least two, and therefore it contains a cycle, a contradiction to the assumption that $H[S]$ is acyclic. Thus, S cannot contain any outer chord, and the theorem follows. □

2.3   NP-completeness for a given tree

The last result of this section is the NP-completeness when the dominating set is restricted to be isomorphic to a *given* tree.

Theorem 4. *Let T be a given tree. Then $\{T\}$-Dominating Set is NP-complete in circle graphs when T is part of the input.*

$I = \{1,1,1,1,2,2,2,2,3\}$

$\{1,2,2\}$ $\{1,2,2\}$ $\{1,1,3\}$

T

G

r

S

$\{1,2,2\}$ $\{1,1,3\}$

$\{1,2,2\}$

(a)   (b)   (c)
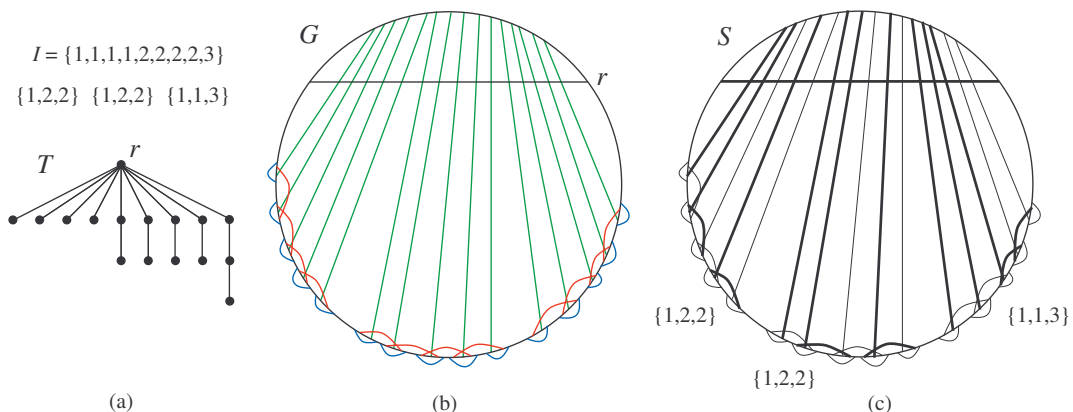
Fig. 7. Reduction in the proof of Theorem 4. (a) Instance I of 3-Partition, with $n = 9$, $m = 3$, and $B = 5$, together with the associated tree T. (b). Circle graph G built from I. (c) The thick chords define a $\{T\}$-dominating set S in G.

Proof: We present a reduction from the 3-Partition problem, which consists in deciding whether a given multiset of $n = 3m$ integers I can be partitioned into m triples that all have the same sum B. The 3-Partition problem is strongly NP-complete, and in addition, it remains NP-complete even when every integer in I is strictly between $B/4$ and $B/2$ [18]. Let $I = \{a_1, \ldots, a_n\}$ be an instance of 3-Partition, in which we can assume that the a_i's are between $B/4$ and $B/2$, and let $B = \sum_{i=1}^n a_i/m$ be the desired sum. Note that we can also assume that B is an integer, as otherwise I is obviously a No-instance.

We proceed to define a tree T and to build a circle graph G that has a $\{T\}$-dominating set S if and only if I is a Yes-instance of 3-Partition. Given $I = \{a_1, \ldots, a_n\}$, let T be the rooted tree obtained from a root r to which we attach a path with a_i vertices, for $i = 1, \ldots, n$; see Fig 7(a) for an example with $n = 9$, $m = 3$, and $B = 5$. (In this figure, for simplicity not all the a_i's are between $B/4$ and $B/2$, but we assume that this fact is true in the proof.) Note that $|V(T)| = mB + 1$.

The circle graph G is obtained as follows; see Fig 7(b) for the construction corresponding to the instance of Fig 7(a): We start with a chord r that will correspond to the root of T. Now we add mB parallel chords $g_1, \ldots, g_{mB}$ intersecting only with r. These chords are called *branch* chords; cf. the green chords in Fig 7(b). We can assume that the endpoints of the branch chords are ordered clockwise in the circle. For $i = 1, \ldots, mB$, we add a chord b_i incident only with g_i. These chords are called *pendant* chords; cf. the blue chords in Fig 7(b), where for better visibility these chords have been depicted outside the circle. Finally, for $i \in \{1, 2, \ldots, mB\} \setminus \{B, 2B, \ldots, mB\}$, we add a chord r_i whose first endpoint is exactly after the first endpoint of b_i (in the anticlockwise order, starting from any of the endpoints of the root r), and whose second endpoint is exactly before the second endpoint of b_{i+1}. These chords are called *chain* chords; cf. the red chords in Fig 7(b). Note that r_i is adjacent to g_i, g_{i+1}, b_i, and b_{i+1}. This completes the construction of the circle graph G. Each one of the m connected components that remain in G after the removal of r and the parallel chords is called a *block*.

Let first I be a Yes-instance of 3-Partition, and we proceed to define a $\{T\}$-dominating set S in G. For $1 \le j \le m$, let $B_j = \{a_1^j, a_2^j, a_3^j\}$ be the j-th triple of the 3-partition of I; in the instance of Fig 7(a), we have $B_1 = \{1, 2, 2\}, B_2 = \{1, 2, 2\}, B_3 = \{1, 1, 3\}$. We include the chord r in S, plus the following chords for each $j \in \{1, \ldots, m\}$: For $i \in \{1, 2, 3\}$, we add to S the branch chord $g_{(j-1)B+\sum_{k=1}^{i-1} a_k^j + 1}$ plus, if $a_i^j \ge 2$, the

chain chords $r_{(j-1)B+\sum_{k=1}^{i-1} a_k^j + \ell}$ for $\ell \in \{1, \ldots, a_i^j - 1\}$; cf. the thick chords in Fig 7(c). It can be easily checked that S is a $\{T\}$-dominating set of G.

Conversely, let S be a $\{T\}$-dominating set S in G, and note that we can assume that the root of T has arbitrarily big degree. As the vertex of G corresponding to the chord r is the only vertex of G of degree more than 6, necessarily r belongs to S, and corresponds to the root of T.

We claim that S contains no pendant chord. Indeed, by construction of G, exactly n of the branch chords are in S, which dominate exactly n pendant chords. As $G[S \setminus \{r\}]$ consists of n disjoint paths, each attached to r through a branch chord, the total number of chords in these paths which are not branch chords is $mB - n$. These $mB - n$ pendant or chain chords must dominate the pendant chords that are not dominated by branch chords, which are also $mB - n$ many. Assume that a pendant chord b belongs to S. Since T is a tree, there must exist a path P in S between b and one of the branch chords, say g. Assume that P contains p chords, including b but not g. It is clear that b is the only pendant chord contained in P, as otherwise P would have a cycle. Therefore, P has p chords and dominates exactly $p - 1$ pendant chords that are not dominated by branch chords, which contradicts the fact that $mB - n$ pendant or chain chords must dominate the $mB - n$ pendant chords that are not dominated by branch chords. Hence, S contains no pendant chord, so S contains exactly $mB - n$ chain chords.

Since T is a tree, each path in S made of consecutive chain chords intersects exactly one branch chord. As the a_i's are strictly between $B/4$ and $B/2$, each block has exactly 3 branch chords in S. The fact that chain chords are missing between consecutive blocks assures the existence of a 3-partition of I. More precisely, the restriction of S to each block defines the integers belonging to each triple of the 3-partition of I as follows. For a branch chord $g_i \in S$, let P_i be the path in S hanging from g_i, which consists only of chain chords. Then, for each branch chord $g_i \in S$, the corresponding integer is defined by the number of vertices in P_i plus one. By the above discussion, these m triples define a 3-partition of I. The theorem follows. $\square$

To conclude this section, it is worth noting here that $\{T\}$-Dominating Set is $W[2]$-hard in general graphs. This can be proved by an easy reduction from Set Cover parameterized by the number of sets, which is $W[2]$-hard [29]. Indeed, let $\mathcal{C}$ be a collection of subsets of a set S, and the question is whether there exist at most k subsets in $\mathcal{C}$ whose union contains all elements of S. We construct a graph G as follows. First, we build a bipartite graph $(A \cup B, E)$, where there is a vertex in A (resp. B) for each subset in $\mathcal{C}$ (resp. element in S), and there is an edge in E between a vertex in A and a vertex in B if the corresponding subset contains the corresponding element. We add a new vertex v, which we join to all the vertices in A, and $k + 1$ new vertices joined only to v. It is then clear that G has a dominating set isomorphic to a star with exactly k leaves if and only if there is a collection of at most k subsets in $\mathcal{C}$ whose union contains all elements of S.

3   Polynomial and FPT algorithms

In this section we provide polynomial and FPT algorithms for finding dominating sets in a circle graph which are isomorphic to trees. Namely, in Section 3.1 we give a polynomial-time algorithm to find a dominating set isomorphic to *some* tree. This algorithm contains the main ideas from which the other algorithms in this section are inspired. In Section 3.2 we modify the algorithm 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 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.

3.1   Polynomial algorithm for trees

Note that, in contrast with Theorem 5 below, Theorem 3 in Section 2.2 states that, if $\mathcal{F}$ is the set of all forests, then $\mathcal{F}$-Dominating Set is $W[1]$-hard in circle graphs. This is one of the interesting examples where the fact of imposing connectivity constraints in a given problem makes it computationally easier, while it is usually not the case (see for instance [4, 30]).

Theorem 5. *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 an arbitrary 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*. Without loss of generality, 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 $a \leq c \leq d \leq b$, 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 chords in it, i.e., 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 *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.*

Proof: All the chords with both endpoints either in $[a, c]$ or in $[d, b]$ are dominated by T, since T is $(ab - cd)$-dominating. Hence we just have to prove that the chords with one endpoint in $[a, c]$ and one in $[d, b]$ are dominated by T. Since T spans $ab - cd$, there are in T a chord γ with endpoint a and a chord γ' with endpoint c. Since T is split by $ab - cd$, there is a unique chord uv in T with one endpoint in $[a, c]$ and one in $[d, b]$. In $T \backslash \{uv\}$, γ and γ' are in the same connected component. Indeed, otherwise their connected components span two disjoint intervals of $[a, c]$. But uv is the unique chord of T with one endpoint in $[a, c]$ and one in $[d, b]$, thus uv cannot connect these components. So, if T is a tree, γ and γ' are in the same connected component.

Thus for each point p in $[a, c]$, there is a chord ef of the connected component of γ and γ' such that $a \leq e \leq p \leq f \leq c$. Therefore, the chords with one endpoint in $[a, c]$ and one endpoint in $[d, b]$ are dominated. $\qquad\square$
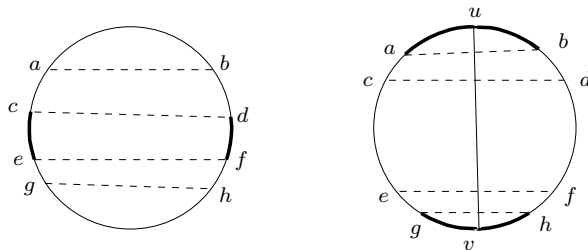
Fig. 8. On the left (resp. right), regions corresponding to Property **T1** (resp. Property **T2**). Full lines correspond to real chords of C, dashed lines correspond to the limit of regions. Bold intervals correspond to intervals with no chord of C with both endpoints in the interval.

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 $a \leq c \leq e \leq g \leq h \leq f \leq d \leq b$. 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. 8).

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 $u \leq a \leq c \leq e \leq g \leq v \leq h \leq f \leq d \leq b$, 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. 8).

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), which will be helpful in the dynamic programming routine. 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.

Claim 8 *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.*

Proof: Let us now consider a chord γ with one endpoint α in $[c, d]$ and one endpoint β in $[d, c]$. First assume that β is in $[b, a]$. Since F is split by $ab - cd$, there is a chord of F with one endpoint in $[a, c]$ and one in $[d, b]$, and such a chord dominates γ. Therefore, by symmetry, we can assume that β is in $[a, c]$. Since F spans $ab - cd$, there is a chord in F with endpoint c. Since F is split by $ab - cd$, there is a chord $\omega = uv$ of F, in the same connected component as the chord with endpoint c, with one endpoint in $[a, c]$ and one endpoint in $[d, b]$. If $\beta \leq u \leq \alpha \leq v$, then the chord γ is dominated by F. Thus we can assume that $u \leq \beta < \alpha$. And note that by assumption, $\beta \leq c \leq \alpha$. But in F, the chord with endpoint c is connected to the chord ω, thus there is a chord wz of F such that $w \leq \beta \leq z \leq \alpha$, and therefore the chord γ is dominated by F, which achieves the proof of the claim. $\square$

Note that the proof of Claim 8 is symmetric, and then the same result is still true for the intervals $[a, b]$ and $[b, a]$. Note also that the same result holds without the asumption that F is $(ab - cd)$-dominating.

Claim 9 *Property **T1** is correct.*

Proof: Let us prove that $F_1 \cup F_2$ is a forest and that it is valid for the region $ab - gh$. For an illustration refer to Fig. 8. Since F_1 and F_2 span $ab - cd$ and $ef - gh$ respectively, all the endpoints of the chords of F_1 are in $[a, c] \cup [d, b]$, and those of F_2 are in $[e, g] \cup [h, f]$. Thus the order $a \leq c \leq e \leq g \leq h \leq f \leq d \leq b$ ensures that a chord of F_1 cannot cross of chord of F_2. Therefore, $F_1 \cup F_2$ is still a forest and the connected components of the union are precisely the connected components of F_1 and the connected components of F_2.

Since F_1 spans $ab - cd$, there is a chord of F_1 with endpoint a and one chord with endpoint b, and the same holds for F_2 and g, h. Then $F_1 \cup F_2$ spans the region $ab - gh$.

Since F_1 is split by $ab - cd$, there is exactly one chord per connected component between $[a, c]$ and $[d, b]$, thus also between $[a, g]$ and $[h, b]$. The same holds for F_2. Thus each connected component of $F_1 \cup F_2$ has exactly one chord with one endpoint in $[a, g]$ and the other one in $[h, b]$. So $F_1 \cup F_2$ is split by $ab - gh$.

Let us now prove that $F_1 \cup F_2$ is $(ab - gh)$-dominating. Let us verify that all the chords in the interval $[a, g]$ are dominated by $F_1 \cup F_2$. By symmetry, the same will hold for the interval $[h, b]$. All the chords with both endpoints in the interval $[a, c]$ are dominated by F_1, and those with both endpoints in the interval $[e, g]$ are dominated by F_2. By assumption, there is no chord in the interval $[c, e]$. The chords with one endpoint in $[c, d]$ and one endpoint in $[d, c]$ are dominated by F_1 by Claim 8, and those with one endpoint in $[e, f]$ and one endpoint in $[f, e]$ are dominated by F_2. Thus all the chords with both endpoints in $[a, c]$ are dominated by $F_1 \cup F_2$, which ensures that $F_1 \cup F_2$ is $(ab - gh)$-dominating.

Therefore, $F_1 \cup F_2$ is valid for the region $ab - gh$. $\qquad\square$

Claim 10 *Property* **T2** *is correct.*

Proof: Let F_1 and F_2 be two valid forests for $ab - cd$ and for $ef - gh$, respectively (F_2 being possibly empty), and let uv be a chord with endpoints u and v, such that $u \leq a \leq c \leq e \leq g \leq v \leq h \leq f \leq d \leq b$ and such that there is no chord with both endpoints in either $[u, a]$, or $[g, v]$, or $[v, h]$, or $[b, u]$. For an illustration, refer also to Fig. 8.

First note that $T = F_1 \cup F_2 \cup \{uv\}$ is a tree. Indeed, as in Claim 9, one can prove that $F_1 \cup F_2$ is a forest with exactly one chord with one endpoint in $[a, g]$ and one endpoint in $[h, b]$ per connected component. Thus, the addition of uv ensures that T is a tree.

Since F_1 and F_2 spans $ab - cd$ and $ef - gh$ respectively, T spans $df - ce$. Indeed, there are chords intersecting d and c in F_1, chords intersecting e and f in F_2, and all the chords are strictly inside $[d, c] \cup [e, f]$. Note that when the forest F_2 is empty, there is a chord intersecting v, and thus the tree T spans $df - ce$.

The tree T spans $df - ce$, since there is exactly one chord with one endpoint in $[d, c]$ and one endpoint in $[e, f]$, which is precisely the chord uv.

Let us prove that T is $(df - ce)$-dominating. First note that the chords with one endpoint in $[u, v]$ and one endpoint in $[v, u]$ are dominated by uv. By symmetry, we just have to prove that the chords with both endpoints in $[u, v]$ are dominated by T. By symmetry again, we just have to prove that the chords with both endpoints in $[u, c]$ are dominated. There is no chord with both endpoints in $[u, a]$, the chords with both endpoints in $[a, c]$ are dominated by F_1, since F_1 is $(ab - cd)$-dominating. By Claim 8, the chords with one endpoint in $[a, b]$ and one endpoint in $[b, a]$ are dominated by F_1, thus the chords with one endpoint in $[u, a]$ and one in $[a, c]$ are dominated.

Therefore, we conclude that T is a valid tree for $df - ce$. $\qquad\square$

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 prove that $v^1_{ab,cd} = v^f_{ab,cd}$ and $v^2_{ab,cd} = v^t_{ab,cd}$, and that Algorithm 1 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 11 *For any region $ab - cd$, $v^1_{ab,cd} = 1$ (resp. $v^2_{ab,cd} = 1$) if and only if $v^f_{ab,cd} = 1$ (resp. $v^t_{ab,cd} = 1$).*

Proof: Let $ab - cd$ be a region such that $v^f_{ab,cd} = 1$. Therefore, there is a set of chords of size one which is valid for $ab - cd$. Let ω be this chord. Since ω spans a, b, c and b, ω has endpoints a, b, c, d. This implies that $a = c$ and $b = d$, i.e., ω is precisely the chord ab. If $v^f_{ab,cd} = 1$, then $a = c$, $b = d$, and the chord ab exists, which corresponds exactly to the initialization of the algorithm. Conversely, it is clear that by definition the chord ab is valid for the region $ab - ab$. Thus, $v^1_{ab,cd} = 1$ if and only if $v^f_{ab,cd} = 1$. The same result holds for trees, since a forest of size one is a tree. $\square$

**Claim 12** *For any region $ab - cd$, $v^f_{ab,cd} \leq v^1_{ab,cd}$ (resp. $v^t_{ab,cd} \leq v^2_{ab,cd}$).*

Proof: The claim is true for the initialization, since if $v^1_{ab,cd} = 1$ then $v^f_{ab,cd} = 1$. By induction it is still true for all integers k, since Properties **T1** and **T2** are correct, and when a value is affected in the dynamic programming of Algorithm 1, one of the two properties is applied. $\square$

Claim 13 *For any region $ab - cd$, $v^t_{ab,cd} \geq v^2_{ab,cd}$ and $v^f_{ab,cd} \geq v^1_{ab,cd}$.*

Proof: Let us prove it by induction on j. Claim 11 ensures that the result is true for $j = 1$. Assume that for all $j < k$, if $v^f_{ab,cd} = j$ then $v^1_{ac,bd} \leq j$ and that if $v^t_{ac,bd} = j$ then $v^2_{ac,bd} \leq j$.

Let us first prove that the induction step holds for trees. We now prove that if $v^t_{ab,cd} = j$, then $v^2_{ab,cd} \leq j$. Let T be a valid tree of size j for the region $ab - cd$. Since T spans $ab - cd$, there is exactly one chord uv with one endpoint in $[a, c]$ and one endpoint in $[d, b]$. Let F_1 be the restriction of T to the chords with both endpoints in $[a, c]$, and let F_2 be the restriction of T to the chords with both endpoints in $[d, b]$. Note that $T = F_1 \cup F_2 \cup \{uv\}$. Let e, f (resp. g, h) be the points of the circle graph intersected by F_1 (resp. F_2) such that $a \leq e \leq u \leq f \leq c$ (resp. $d \leq h \leq v \leq g \leq b$), and e, f (resp. g, h) are as near as possible from u (resp. v) (see Fig. 9 for an example). Let us denote by j_1 (resp. j_2) the size of F_1 (resp. F_2). Note that $j_1 + j_2 = j - 1$.
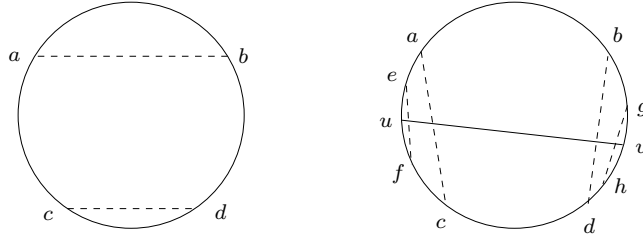
Fig. 9. On the left the original tree, and on the right the partition of the tree into two forests and a chord uv.

Let us prove that F_1 is valid for $ac - ef$, that F_2 is valid for $db - hg$, and that Property **T2** can be applied to F_1, F_2, and the chord uv. By symmetry, we just have to prove that F_1 is valid. There are chords intersecting e, f by definition of e, f, and chords intersecting a, c since T spans $ab - cd$. The endpoints of the chords are in $[a, e] \cup [f, c]$, since T spans $ab - cd$ and e, f are the nearest points from u which are in T. Thus F_1 spans $ac - ef$.

If there is a connected component of F_1 with no chord from $[a, e]$ to $[f, c]$, then $F_1 \cup F_2 \cup \{uv\}$ cannot be a tree, since it would not be connected. If a connected component of F_1 has two chords from $[a, e]$ to $[f, c]$, then $F_1 \cup \{u, v\}$ has a cycle, which contradicts the fact that T is a tree. Thus F_1 spans $ac - ef$.

Let us prove that F_1 is $(ac - ef)$-dominating. Indeed, if there is a chord with both endpoints in $[a, e]$ which is not dominated by F_1, it cannot be dominated by F_2 and uv, since none of their endpoints is in this interval. Thus T is not valid. Hence all the chords in the interval $[a, e]$, and by symmetry also in the interval $[f, c]$, are dominated by F_1. Therefore F_1 is valid. By induction hypothesis, since the size of F_1 is at most j_1, we have $v^2_{ac,ef} \leq j_1$, and the same holds for F_2.

Since T is valid, one can note that there is no chord with both endpoint either in $[e, u]$, or $[u, f]$, or $[h, v]$, or $[v, g]$. Thus Property **T2** can be safely applied and then $v^2_{ab,cd} \leq j$, as we wanted to prove.

Let us now prove that the induction step also holds for forests. Let us now consider a forest F for the region $ab - cd$. If the forest has exactly one connected component, that is, if it is a tree, then the inequality holds by the first part of the induction.

Assume now that F has at least two connected components. Let us now prove that if $v^f_{ab,cd} = j$ then $v^1_{ab,cd} \leq j$. Let $ab - cd$ be a region with $v^f_{ac,bd} = k$. This means that there exists a forest F with one endpoint in a, b, c, d, since F spans $ab - cd$. Since $v^t_{ab,cd} \neq j$,
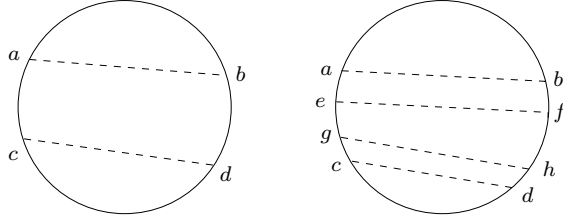
Fig. 10. On the left the original forest, and on the right the partition of the forest into the two forests.

F has at least two connected components. Note that the case when $v^t_{ab,cd} = j$ is treated just above.

Since F spans $ab - cd$, all the endpoints of F are in $[a, c] \cup [d, c]$. Let F_1 be the connected component of the chord with endpoint a. The point e (resp. f) is the point of $[a, c]$ (resp. $[d, b]$) with an endpoint in F_1, and such that there is no endpoint of F_1 after e (resp. before f) in $[a, c]$ (resp. $[d, b]$). Let g (resp. h) be the first endpoint of F after e in $[a, c]$ (resp. before f in $[d, b]$).

Let us denote by F_2 the set $F \setminus F_1$ (see Fig. 10 for an example). Let us prove that F_1 and F_2 are valid for $ab - ef$ and for $gh - cd$, respectively. Since F is $(ab - cd)$-dominating, all the chords with both endpoints either in $[a, e]$ or in $[f, b]$ (resp. $[g, c]$ or $[h, d]$) are dominated by F, thus by F_1 (resp. F_2). Therefore F_1 (resp. F_2) is $(ab - ef)$-dominating (resp. $(gh - cd)$-dominating). Thus, by induction hypothesis we have $v^1_{ac,ef} = v^f_{ac,ef}$ and $v^1_{gh,cd} = v^f_{gh,cd}$. And since Property **T1** can be applied for $F_1 \cup F_2$, by the safeness of Property **T1**, in Algorithm 1 we have $v^1_{ab,cd} \leq v^f_{ab,cd}$. $\square$

Claims 12 and 13 together ensure that $v^f_{ab,cd} = v^1_{ab,cd}$ and that $v^t_{ab,cd} = v^2_{ab,cd}$. Hence, by dynamic programming all the regions of a given size can be found in polynomial time. Let us now explain how we can verify if there is a dominating set isomorphic to some tree of a given size k. This in particular will prove Theorem 5.

Claim 14 *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 $[c, d]$.*

Proof: Assume that there is a region $ab - cd$ with $v^t_{ab,cd} \leq k$ and no chord strictly contained in $[b, a]$ nor in $[c, d]$. Then by Claim 8, all the chords with one endpoint in $[b, a]$ and one endpoint in $[a, b]$ are dominated, and the same holds for the couple c, d. All the chords with both endpoints in $[a, c] \cup [d, b]$ are dominated by Claim 7, as T is valid for $ab - cd$. Since by assumption there is no chord strictly contained in $[b, a]$ and in $[c, d]$, all the chords of the circle graph C are dominated, as by assumption there is no chord in the other intervals. Thus, T is a dominating set.

Conversely, let T be a dominating tree of size k. Let uv be a chord of T which disconnects T. Thus $T \setminus \{uv\}$ has at least two connected components F_1 and F_2. Let a, c be the two extremities of the first one, and let b, d be the extremities of the other ones (see Fig. 9 for an illustration). Let us prove that $v^t_{ab,cd} \leq k$. Indeed, T is a tree by assumption, by definition it spans a, b, c, d, it spans $ab - cd$ since the chord uv is the unique chord from $[a, c]$ to $[d, b]$, and it is $(ab - cd)$-dominating since T is a dominating tree. In addition, since T is a dominating tree, there is no chord with both endpoints either in the interval $[b, a]$ or in $[c, d]$, as otherwise such a chord would not be dominated by T. $\square$

By dynamic programming, Algorithm 1 computes in polynomial time the regions for which there is a valid tree of any size from 1 to n. Given a region $ab - cd$ with $v_{ab,cd}^t \leq k$, 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. This completes the proof of Theorem 5. $\qquad\square$

As Algorithm 1 computes the regions for which there is a valid tree of any size from 1 to n, it can be slightly modified to obtain the following corollary.

Corollary 2. *Let $\mathcal{T}_k$ be the set of all trees of size exactly k. Then $\mathcal{T}_k$-Dominating Set can be solved in polynomial time in circle graphs.*

3.2   FPT algorithm for a given tree

It turns out that when we seek a dominating set isomorphic to a given *fixed* tree T, the problem is FPT parameterized by $|V(T)|$. In order to express the running time of our algorithm, and to prove that it is *subexponential* in $|V(T)|$, we need some definitions. Let T be a tree, and let us root T at an arbitrary vertex r. Let v be a vertex of T. We denote by $T[v]$ the subtree of T induced by v and the descendants of v in the rooted tree. Let $v_1, \ldots, v_l$ be the children of v in the tree T rooted at r. We define $F(v)$ as the forest $T[v_1] \cup T[v_2] \ldots \cup T[v_l]$, which we consider as a multiset with elements $T[v_1], \ldots T[v_l]$, where we consider two isomorphic trees $T[v_i]$ and $T[v_j]$ as the same element. Suppose that $F(v)$ contains exactly s non-isomorphic trees $T_1, \ldots, T_s$, and that for $1 \leq i \leq s$, there are exactly d_i trees in $F(v)$ which are isomorphic to T_i (note that $\sum_{i=1}^s d_i = l$). We define the following parameter, which corresponds to the number of non-isomorphic sub(multi)sets of the multiset $F(v)$:

$$\alpha_r^T(v) \;=\; \prod_{i=1}^s (d_i + 1).$$

Finally, we also define the following three parameters:

$$\alpha_r^T = \max_{v \in V(T)} \alpha_r^T(v)$$

$$\alpha^T = \max_{r \text{ root of } T} \alpha_r^T$$

$$\alpha^t = \max_{T : |V(T)| = t} \alpha^T.$$

Let $t = |V(T)|$. Note that for any tree T, we easily have that $\alpha^T \leq 2^t$, and that if T has maximum degree at most Δ, then it holds that $\alpha^T \leq t \cdot 2^{\Delta - 1}$ (by choosing r to be a leaf of T). In particular, if T is a path on t vertices, is holds that $\alpha^T \leq 2t$. In the following proposition we upper-bound the parameter α^t, seen as a function of t, which will allow us to prove that the running time of the algorithm in Theorem 6 is subexponential.

Proposition 1. $\alpha^t \;=\; 2^{\mathcal{O}(t \cdot \frac{\log \log t}{\log t})} \;=\; 2^{o(t)}$.

Proof: Let t be an integer and let T be a tree which maximizes α^t, i.e., a tree T for which $\alpha^T = \alpha^t$. Let r be a root of T maximizing α_r^T, and let v be a vertex of T such that v maximizes $\alpha_r^T(v)$. We claim that we can assume that $v = r$. Note that if $v \neq r$, then it holds that $\alpha_v^T(v) > \alpha_r^T(v)$, as on the left-hand side we have one more child of v

that contributes to $\alpha_v^T(v)$. Assume for contradiction that $\alpha_r^T(v) > \alpha_r^T(r)$. Then, by the previous inequality it holds that $\alpha_v^T(v) > \alpha_r^T(v) > \alpha_r^T(r)$, contradicting the choice of r. Therefore, we assume henceforth that $v = r$.

Let $T_1, T_2, ..., T_s$ be all the non-isomorphic rooted trees of $F(r)$ sorted by increasing size (where size means number of vertices). As defined before, we denote by d_i the number of occurrences of T_i in $F(r)$. By simplicity in the sequel, let us denote by $k+1$ the size of T_s. We first want to find an upper bound on s.

We will need the fact that the number of unlabeled rooted trees of size ℓ is asymptotically equal to $a \cdot d^\ell \ell^{-3/2}$, where $a \simeq 0.4399$ and $d \simeq 2.9558$ (see for instance [15, Chapter VII.5]). It follows that there exist two constants c, c' and a constant d such that for all ℓ, the number $N_t(\ell)$ of unlabeled rooted trees of size ℓ satisfies

$$c \cdot d^\ell \ell^{-3/2} \ \leq \ N_t(\ell) \ \leq \ c' \cdot d^\ell \ell^{-3/2}. \tag{1}$$

Claim 15 *There exists a constant c such that $s \leq ct/\log t$.*

Proof: Let us first prove by contradiction that all the trees of size at most k appear in $F(v)$. Assume that there is a tree T_a of size at most k which is not in $F(v)$. Let T' be the same tree as T, rooted at r', except that we replace all the occurrences of T_s in $F(r')$ by occurrences of T_a. Since the size of T_a is at most k, T' contains strictly less vertices, say l, than T. Thus in order to have t vertices, we attach to the root of T' l new trees isomorphic to the singleton-tree. Note that T' has also size t. Let us calculate the difference between $\alpha_r^T(r)$ and $\alpha_{r'}^{T'}(r')$. Note that by maximality of T, we have $\alpha_r^T(r) \geq \alpha_{r'}^{T'}(r')$. For all $1 \leq i \leq s-1$ such that T_i is not the singleton-tree, by construction the number of trees isomorphic to T_i in $F(r)$ in T is equal to the number of occurrences of T_i in $F(r')$ in T'. Thus $d_i = d_i'$ for all $1 \leq i \leq s-1$ when T_i is not the singleton-tree. Since, by construction, there are the same number of occurrences of T_s in $F(r)$ and of T_a in $F(r')$, we have $d_s = d_a$. Thus the only difference between $\alpha_r^T(r)$ and $\alpha_{r'}^{T'}(r')$ is the term corresponding to the singleton-tree. If the number of occurrences of the singleton-tree in $F(r)$ is d^*, then the number of occurrences of the singleton-tree in $F(r')$ is $d^* + l$. Thus $\alpha_{r'}^{T'}(r') > \alpha_r^T(r)$, which contradicts the maximality of T.

Thus, we can assume that all the non-isomorphic rooted trees of size at most k appear in $F(r)$. By Equation (1), there are at least $c \cdot d^k k^{-3/2}$ non-isomorphic rooted trees of size k. Therefore, if there is a tree of size $k+1$ in $F(r)$, then necessarily

$$|V(T)| \ = \ t \ \geq \ k \cdot c' \cdot d^k k^{-3/2} \ = \ c \cdot d^k/\sqrt{k}.$$

Note that, in particular, we have that $k \geq \log t$ for t large enough. Indeed, when we replace k by $\log t$, the inequality is satisfied. In the following, when we write $\log$ we mean $\log_d$.

Since one can easily check that the number of rooted trees of size at most $k+1$ is less that the number of rooted trees of size exactly $k+2$, and since T_s has size $k+1$, by the previous inequalities we have

$$s \ \leq \ c \cdot d^{k+2} \cdot (k+2)^{-3/2} \ \leq \ c \cdot d^2 \cdot d^k/k^{3/2} \ \leq \ c_1/k \cdot d^k/\sqrt{k} \ \leq \ c_2 t/k \ \leq \ c_2 t/\log t,$$

where the last inequality is a consequence of the fact that $k \geq \log t$. Thus, there exists a constant, called again c for simplicity, such that $s \leq ct/\log t$, as we wanted to prove. $\square$

We now state a useful claim.

Claim 16 *Let $x_1, \ldots, x_k$ be some real variables and let P be the polynomial such that $P(x_1, \ldots, x_k) = \prod_{i=1}^k x_i$. Under the constraint $\sum_{i=1}^k x_i = \ell$, the polynomial is maximized when $x_i = \ell/k$ for all $1 \leq i \leq k$.*

Proof: Assume for contradiction that this is not the case. Then by symmetry we can assume that $x_1 > \ell/k$. Thus there exists another value, say x_2, such that $x_2 < \ell/k$. Let $\varepsilon = \min\{x_1 - \ell/k, \ell/k - x_2\}$. To compare the two values of the polynomial, we just have to compare the product $x_1 \cdot x_2$. One can easily verify that $x_1 \cdot x_2 < (x_1 - \varepsilon)(x_2 + \varepsilon)$, contradicting the fact that the polynomial was maximized. □

Claim 16 ensures that α^T is maximized when all the d_i's are equal. Since T contains t vertices and since each tree contains at least one vertex, it holds that $\sum_{i=1}^{s} d_i \le t$. Thus the function α^T is maximized when we have $d_i = t/s$ for all $1 \le i \le k$, i.e.,

$$\alpha^T \;\le\; \max_{1 \le s \le ct/\log t} \prod_{i=1}^{s}(d_i + 1) \;\le\; \max_{1 \le s \le ct/\log t} (t/s + 1)^s.$$

Claim 17 *Let t be a large enough integer. The real function $f_t : x \mapsto (t/x + 1)^x$ is increasing in the interval $[1, ct/\log t]$.*

Proof: The derivative of the function f_t is the following

$$f_t'(x) = \exp(x \log(t/x + 1)) \cdot (\log(t/x + 1) - 1/(1 + t/x)).$$

Note that the first term is always a positive function. We also have $1/(1 + t/x) \le 1$. Thus we have $\log(t/x + 1) - 1/(1 + t/x) \ge \log(t/x + 1) - 1 \ge 0$ when t is large enough, since $x \in [1, ct/\log(t)]$. □

Since f_t is an increasing function by Claim 17, and since $s \le ct/\log t$ by Claim 15, we have that

$$\alpha^T \;\le\; (\log t/c + 1)^{ct/\log t} \;\le\; 2^{c' \cdot t \cdot \frac{\log \log t}{\log t}} \;,$$

for some constant c', which completes the proof of Proposition 1. □

We are now ready to state Theorem 6, which should be compared to Theorem 4 in Section 2.3. We use Proposition 1 to conclude that the running time is subexponential.

Theorem 6. *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 $\mathcal{O}(\alpha^T \cdot n^{\mathcal{O}(1)}) = 2^{o(t)} \cdot n^{\mathcal{O}(1)}$. In particular, if T has bounded degree, $\{T\}$-Dominating Set can be solved in polynomial time in circle graphs.*

Proof: The idea of the proof is basically the same as in the proof of Theorem 5. The main difference is that in the proof of Theorem 5, when Properties **T1** or **T2** are satisfied, we can directly apply them and still obtain a forest or a tree. In the current proof, when we make the union of two forests, we have to make sure that the union of the 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 the two properties stated below, whenever it is possible to create forests which are induced by the children of the same vertex of T. Let us first give some intuition on the algorithm.

In the following we consider the tree T rooted at an arbitrary vertex r. Let $w_1, \ldots, w_l$ be some vertices of T which are children of the same vertex y. The *subforest of T induced by $w_1, \ldots, w_l$*, denoted by $F(w_1, \ldots, w_l)$, is the forest $T[w_1] \cup T[w_2] \ldots \cup T[w_l]$.

[See Daniel's remarks] 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 v,

which define 2^k possible types of subforests in $F(v)$. But the key point 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, which corresponds to the parameter $d_i + 1$ (as we may not choose any copy of it) defined before the statement of Proposition 1. Note that carrying out this partition into isomorphism classes can be done in polynomial time (in 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].

Therefore, if we proceed in this way, the number of such subforests for each vertex v is at most $\alpha_r^T(v)$. As we repeat this procedure for every node of T, the cost of this routine per vertex is at most $\alpha_r^T = \max_{\{v \in V(T)\}} \alpha_r^T(v)$. And as we chose the root arbitrarily, it follows that the function can be upper-bounded by $\alpha^T = \max_{\{r \text{ root of } T\}} \alpha_r^T$, which in turn can be upper-bounded by $\alpha^t = \max_{\{T:|V(T)|=t\}} \alpha^T$, which is a subexponential function by Proposition 1. We would like to note that this step is the unique non-polynomial part of the algorithm.

Let us now explain more precisely the outline of the algorithm. An induced subtree T_1 of the input circle graph is *valid* for a region $ab - cd$ and a tree $T[w]$, if it is valid for $ab - cd$, and if, in addition, there is an isomorphism between T_1 and $T[w]$ for which the unique chord between $[a, c]$ and $[b, d]$ corresponds to w. A forest F_1 is *valid* for a region $ab - cd$ and $F(w_1, \dots, w_l)$, if it is valid for $ab - cd$, and if there is an isomorphism between F_1 and $F(w_1, \dots, w_l)$ for which the unique chord between $[a, c]$ and $[b, d]$ of each connected component $T[w_i]$ corresponds to vertex w_i. Let us now state the two properties that correspond to Properties **T1** and **T2** of Theorem 5.

F1 Let F_1 and F_2 be two valid forests for $ab - cd$ and $F(v_1, \dots, v_l)$, and for $ef - gh$ and $F(w_1, \dots, w_m)$, respectively. Assume in addition that $a \leq c \leq e \leq g \leq h \leq f \leq d \leq b$. Assume also that, for all $1 \leq i \leq l$, $1 \leq j \leq m$, the vertices v_i and w_j are pairwise distinct and are children of the same vertex y of T. 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 $ab - gh$ and $F(v_1, \dots, v_l, w_1, \dots, w_m)$.

F2 Let F_1 and F_2 be two valid forests for $ab - cd$ and $F(v_1, \dots, v_l)$, and for $ef - gh$ and $F(w_1, \dots, w_m)$, respectively (F_2 being possibly empty), and let uv be a chord of the input graph C. Assume that $u \leq a \leq c \leq e \leq g \leq v \leq h \leq f \leq d \leq b$ and that there is no chord with both endpoints either in $[u, a]$, or in $[g, v]$, or in $[v, h]$, or in $[b, u]$. Assume also that there exists a vertex y of T with exactly $l + m$ children $v_1, \dots, v_l, w_1, \dots, w_m$ Then $F_1 \cup F_2 \cup \{uv\}$ is a tree which is valid for $df - ce$ and $T[y]$. When F_2 is empty, we consider that e, f, g, h correspond to the point v.

In the proof of Theorem 5, we have seen that the validity of the corresponding regions is satisfied. Thus, we just have to verify that the tree or the forest which is created is isomorphic to the target tree T, and that the chords with one endpoint in each side are children of the same vertex. The union of the two isomorphisms, and the fact that the chords with one endpoint in both sides are the children of y, ensures that both properties are true. Indeed, for example for Property **F2**, since the chords with one endpoint in each interval are exactly the children of y, it holds that the chord corresponding to the vertex y intersects exactly its children.

For each region $ab - cd$ and each tree $T[w]$, we define a boolean variable $b_{ab,cd,w}^t$ [We don't need this variable], which is set to 'true' if and only if there is a valid tree for $ab - cd$ and $T[w]$. For each region $ab - cd$ and each forest $F(w_1, \dots, w_l)$, we define a boolean variable $b_{ab,cd,w_1,\dots,w_l}^f$ which is set to true if and only if there is a valid forest for $ab - cd$ and $F(w_1, \dots, w_l)$.

By a dynamic programming similar to Algorithm 1 in Theorem 5, we can compute all the regions $ab - cd$ and all vertices v of T for which $b^t_{ab,cd,v} = $ true (and the same for forests). If there is a region $ab - cd$ for which $b^t_{ab,cd,r} = $ true, and such that there is no chord with both endpoints either in $[b, a]$ or in $[c, d]$, then the tree T dominates all the chords in the input circle graph C. Indeed, the safeness of Properties **F1** and **F2** ensures that there is a valid tree isomorphic to T for the region $ab - cd$. And Claim 14 in the proof of Theorem 5 ensures that this tree is indeed a dominating tree.

Note that, indeed, the unique non-polynomial step of the algorithm consists in generating the collection of non-isomorphic subforests, which are at most α^T many. Thus, the dynamic programming algorithm runs in time $\mathcal{O}(\alpha^T \cdot n^{\mathcal{O}(1)})$. Again, we did not make any effort to optimize the degree of the polynomial in the running time. □

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

References

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
2. J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed Parameter Algorithms for Dominated Set and Related Problems on Planar Graphs. *Algorithmica*, 33(4):461–493, 2002.
3. N. Alon and S. Gutner. Kernels for the Dominating Set Problem on Graphs with an Excluded Minor. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(066), 2008.
4. O. Amini, D. Peleg, S. Pérennes, I. Sau, and S. Saurabh. Degree-Constrained Subgraph Problems: Hardness and Approximation. In *Proc. of the 6th Workshop on Approximation and On-line Algorithms (ALGO/WAOA)*, volume 5426 of *LNCS*, pages 29–42, 2008.
5. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.
6. B. Courcelle. The monadic second-order logic of graphs: definable sets of finite graphs. In *In Proc. of the 14th International Workshop on Graph-theoretic Concepts in Computer Science (WG)*, number 344 in LNCS, pages 30–53, 1988.
7. M. Cygan, G. Philip, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Dominating set is fixed parameter tractable in claw-free graphs. *Theoretical Computer Science*, 412(50):6982–7000, 2011.
8. P. Damaschke. The Hamiltonian Circuit Problem for Circle Graphs is NP-Complete. *Information Processing Letters*, 32(1):1–2, 1989.
9. M. Damian-Iordache and S. V. Pemmaraju. Hardness of approximating independent domination in circle graphs. In *10th International Symposium on Algorithms and Computation (ISAAC)*, volume 1741 of *LNCS*, pages 56–69, 1999.
10. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.
11. E. S. Elmallah and L. K. Stewart. Independence and domination in polygon graphs. *Discrete Applied Mathematics*, 44(1-3):65–77, 1993.
12. S. Even and A. Itai. Queues, stacks and graphs. In A. Press, editor, *Theory of Machines and Computations*, pages 71–86, 1971.
13. M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.
14. P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.
15. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer Verlag, 2006.
16. F. Fomin, S. Gaspers, P. Golovach, K. Suchan, S. Szeider, E. Jan Van Leeuwen, M. Vatshelle, and Y. Villanger. k-Gap interval Graphs. To appear in the *Proc.*

of the 10th Latin American Theoretical INformatics Symposium (LATIN), available at `http://arxiv.org/abs/1112.3244`, 2012.

17. M. Garey and D. Johnson. *Computers and Intractability*. W.H. Freeman, San Francisco, 1979.
18. F. Gavril. Algorithms for a maximum clique and a maximum independent set of a circle graph. *Networks*, 3:261–273, 1973.
19. F. Gavril. Minimum weight feedback vertex sets in circle graphs. *Information Processing Letters*, 107(1):1–6, 2008.
20. E. Gioan, C. Paul, M. Tedder, and D. Corneil. Circle Graph Recognition in Time $O(n + m) \cdot \alpha(n + m)$. Manuscript available at `http://arxiv.org/abs/1104.3284`, 2011.
21. S. M. Hedetniemi, S. T. Hedetniemi, and D. F. Rall. Acyclic domination. *Discrete Mathematics*, 222(1-3):151–165, 2000.
22. M. Jiang and Y. Zhang. Parameterized Complexity in Multiple-Interval Graphs: Domination. In *Proc. of the 6th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 7112 of *LNCS*, pages 27–40, 2011.
23. J. M. Keil. The complexity of domination problems in circle graphs. *Discrete Applied Mathematics*, 42(1):51–63, 1993.
24. J. M. Keil and L. Stewart. Approximating the minimum clique cover and other hard problems in subtree filament graphs. *Discrete Applied Mathematics*, 154(14):1983–1995, 2006.
25. T. Kloks. Treewidth of circle graphs. *International Journal of Foundations of Computer Science*, 7(2):111–120, 1996.
26. D. Marx. Parameterized complexity of independence and domination on geometric graphs. In *Proc. of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 4169 of *LNCS*, pages 154–165, 2006.
27. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
28. A. Paz and S. Moran. Nondeterministic polynomial optimization problems and their approximations. *Theoretical Computer Science*, 15:251–277, 1981.
29. J. Rué, I. Sau, and D. M. Thilikos. Dynamic Programming for Graphs on Surfaces. In *Proc. of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 6198 of *LNCS*, pages 372–383, 2010.
30. N. A. Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Press, 1992.
31. J. Spinrad. Recognition of circle graphs. *Journal of Algorithms*, 16(2):264–282, 1994.
32. W. Unger. On the k-Colouring of Circle-Graphs. In *Proc. of the 5th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 294 of *LNCS*, pages 61–72, 1988.
33. W. Unger. The Complexity of Colouring Circle Graphs (Extended Abstract). In *Proc. of the 9th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 577 of *LNCS*, pages 389–400, 1992.
34. G. Xu, L. Kang, and E. Shan. Acyclic domination on bipartite permutation graphs. *Information Processing Letters*, 99(4):139–144, 2006.