



The 4-Steiner Root Problem

Guillaume Ducoffe

► To cite this version:

Guillaume Ducoffe. The 4-Steiner Root Problem. 45th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2019), Jun 2019, Vall de Nuria, Spain. pp.14-26, 10.1007/978-3-030-30786-8_2 . hal-02290671

HAL Id: hal-02290671

<https://hal.science/hal-02290671>

Submitted on 17 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The 4-Steiner Root Problem [★]

Guillaume Ducoffe^{1,2,3}

¹ University of Bucharest, Faculty of Mathematics and Computer Science, Romania

² The Research Institute of the University of Bucharest ICUB, Romania

³ National Institute for Research and Development in Informatics, Romania

Abstract. The k^{th} -power of a graph G is obtained by adding an edge between every two distinct vertices at a distance $\leq k$ in G . We call G a k -Steiner power if it is an induced subgraph of the k^{th} -power of some tree T . In particular, G is a k -leaf power if all vertices in $V(G)$ are leaf-nodes of T . Our main contribution is a polynomial-time recognition algorithm of 4-Steiner powers, thereby extending the decade-year-old results of (Lin, Kearney and Jiang, *ISAAC'00*) for $k = 1, 2$ and (Chang and Ko, *WG'07*) for $k = 3$. As a byproduct, we give the first known polynomial-time recognition algorithm for 6-leaf powers. Our work combines several new algorithmic ideas that help us overcome the previous limitations on the usual dynamic programming approach for these problems.

Keywords: k -Leaf Powers; k -Steiner Powers; Clique-tree; Clique-arrangement; Dynamic programming; Maximum matching.

1 Introduction

A basic problem in computational biology is, given some set of species and a dissimilarity measure in order to compare them, find a *phylogenetic tree* that explains their respective evolution. Namely, such a rooted tree starts from a common ancestor and branches every time there is a separation between at least two of the species we consider. In the end, the leaves of the phylogenetic tree should exactly represent our given set of species. We study a related problem that has attracted some attention in Graph theory:

Problem 1 (k -LEAF POWER).

Input: a graph $G = (V, E)$.

Question: Is there a tree T whose leaf-nodes are the vertices in V and such that $uv \in E \iff dist_T(u, v) \leq k$?

The yes-instances of Problem 1 are called k -leaf powers. Their structural and algorithmic properties have been intensively studied (*e.g.*, see [1,8,12] and the

[★] This work was supported by an ICUB Fellowship for Young Researchers and a grant of Romanian Ministry of Research and Innovation CCCDI-UEFISCDI. project no. 17PCCDI/2018.

papers cited therein). However, the complexity of k -LEAF POWER is a longstanding open problem. Very recently, parameterized (FPT) algorithms were proposed for every fixed k on the graphs with degeneracy at most d , where the parameter is $k + d$ [11]. For general graphs, polynomial-time recognition algorithms are known only for $k \leq 5$ [4,6,9]. Characterizations are known only for $k \leq 4$. Several variations of k -leaf powers were introduced in the literature [5,7,9,13]. In this work, we consider k -Steiner powers: a natural relaxation of k -leaf powers where the vertices in the graph may also be internal nodes in the tree T . Interestingly, there is a linear-time reduction from k -LEAF POWER to $(k - 2)$ -STEINER POWER [6]. However, there only exist polynomial-time recognition algorithms for k -Steiner powers for $k \leq 3$ [9,13].

Our Results. We obtain the first improvement on the recognition of k -Steiner powers in a decade, by solving the case $k = 4$. Combining our main result with the aforementioned reduction from k -LEAF POWER to $(k - 2)$ -STEINER POWER [6], we also improve the state of the art for k -leaf powers.

Theorem 1. *There is a polynomial-time algorithm for the problems 4-STEINER POWER and 6-LEAF POWER. For the yes-instances, this algorithm also outputs a corresponding tree T .*

Proving this above Theorem 1, while it may look like a modest improvement in our understanding of the k -LEAF POWER and k -STEINER POWER problems, was technically challenging. In the full version of this paper, we further discuss why the dominant approach for k -STEINER POWER, based on dynamic programming, was already showing its limitations with 4-STEINER POWER. We so believe that one of the main merits of our work is to bring several new ideas in order to tackle with these aforementioned limitations. As such, we expect further uses of these ideas in the study of k -leaf powers and their relatives.

Organization of the paper. We refer to Sec. 2 for any missing definition in what follows. As our starting point we restrict our study to *chordal graphs* and *strongly chordal graphs*, that are two well-known classes in algorithmic graph theory of which k -Steiner powers form a particular subclass [1]. Doing so, we can use various properties of these classes of graphs, such as: the existence of a tree-like representation of chordal graphs, that is called a *clique-tree* [2] and is commonly used in the design of dynamic programming algorithms on this class of graphs; and an auxiliary data structure which is called “*clique arrangement*” and is polynomial-time computable on strongly chordal graphs [16]. Roughly, this clique arrangement encodes all possible intersections of a subset of maximal cliques in a graph. It is worth noticing that clique arrangements were introduced in the same paper as leaf powers, under the different name of “clique graph” [17].

Given a k -Steiner power G , let us call k -Steiner root a corresponding tree T . In Sec. 3 we present new results on the structure of k -Steiner roots that we use in the analysis of our algorithm. Specifically, we show in Sec. 3.1 that in any k -Steiner root T of a graph G , any intersection of maximal cliques in G must

be contained in a particular subtree where no other vertex of G can be present. Furthermore, the inclusion relationships between these “clique-intersections” in G are somewhat reflected by the diameter of their corresponding subtrees in T . This extends prior results from [9,17]. Then, we focus in Sec. 3.2 on the case $k = 4$. For every clique-intersection X in a chordal graph G , we classify the vertices in X into two main categories: “free” and “constrained”, that depend on the other clique-intersections these vertices are contained in. Our study shows that free vertices cause a combinatorial explosion of the number of partial solutions we should store in a naive dynamic programming algorithm. However, we overcome this issue by proving that there always exists a “well-structured” 4-Steiner root where such free vertices are leaves with very special properties.

Sec. 4, 5, 6 and 7 are devoted to the main steps of the algorithm. We start by presenting a constructive proof of a rooted clique-tree with quite constrained properties in Sec. 4. Roughly we carefully control the ancestor/descendant relationships between the edges that are labelled by different minimal separators of the graph. These technicalities are the cornerstone of our approach in Sec. 6 in order to bound the number of partial solutions that we should store in our dynamic programming algorithm.

Then given our special rooted clique-tree T_G , we recall that the maximal cliques and the minimal separators of G can be mapped to the nodes and edges of T_G , respectively. For every node and edge in T_G , we consider the corresponding clique-intersection in G and we precompute by dynamic programming all possible subtrees to which it could be mapped in some well-structured 4-Steiner root of G . Of particular importance is Sec. 5.1 where for any minimal separator S , we give a polynomial-time algorithm in order to generate all the candidate smallest subtrees that could contain S in a well-structured 4-Steiner root of G . The result is then easily extended to the maximal cliques that appear as leaves in our clique-tree (Sec. 5.2). Correctness of these two first parts follows from Sec. 3.2. Finally, in Sec. 5.3 we give a more complicated representation of a family of candidate subtrees $T\langle K_i \rangle$ for all the other maximal cliques K_i . This part is based on a careful analysis of clique-intersections in K_i and several additional tricks. Roughly, our representation in Sec. 5.3 is composed of partially constructed subtrees and of “problematic” subsets that need to be inserted to these subtrees in order to complete the construction. The exact way these insertions must be done is postponed until the very end of the algorithm (Sec. 7).

Sec. 6 is devoted to the encoding of partial solutions in our dynamic programming. Specifically, instead of computing partial solutions at each node of the clique-tree and storing their encodings, we rather pre-compute a polynomial-size subset of *allowed* encodings for each node. Then, the problem becomes to decide whether given such an encoding, there exists a corresponding partial solution. We formalize our approach by introducing an intermediate problem where the goal is to compute a 4-Steiner root with additional constraints on its structure and the distances between some sets of nodes. Finally, we detail in Sec. 7 the resolution of our intermediate problem, thereby completing the presentation of our algorithm. An all new contribution in this part is a greedy procedure,

based on MAXIMUM-WEIGHT MATCHING, in order to ensure some distances' constraints are satisfied by the solutions we generate during the algorithm.

This is only an extended abstract. Full proofs can be found in our technical report [10]. Due to their intricacy we gave up optimizing the running-time of our algorithm. A very rough upper bound would be $\mathcal{O}(n^{16}m^5)$ -time.

2 Preliminaries

For standard graph terminology, see [3]. All graphs in this study are finite, simple, unweighted and connected. Given a graph $G = (V, E)$, let $n := |V|$ and $m := |E|$. The neighbourhood of a vertex $v \in V$ is defined as $N_G(v) := \{u \in V \mid uv \in E\}$. By extension, we define the neighbourhood of a set $S \subseteq V$ as $N_G(S) := (\bigcup_{v \in S} N_G(v)) \setminus S$. The subgraph induced by any subset $U \subseteq V$ is denoted by $G[U]$. For every $u, v \in V$, we denote by $\text{dist}_G(u, v)$ the minimum length (number of edges) of a uv -path. The eccentricity of vertex v is defined as $\text{ecc}_G(v) := \max_{u \in V} \text{dist}_G(u, v)$. The radius and the diameter of G are defined, respectively, as $\text{rad}(G) := \min_{v \in V} \text{ecc}_G(v)$ and $\text{diam}(G) := \max_{v \in V} \text{ecc}_G(v)$. We denote by $\mathcal{C}(G)$ the center of G , *a.k.a.* the vertices with minimum eccentricity.

Steiner roots. The k^{th} -power of G , denoted G^k has same vertex-set V as G and edge-set $\{uv \mid 0 < \text{dist}_G(u, v) \leq k\}$. If there is some tree T such that G is an induced subgraph of T^k then, we call G a k -Steiner power and T a k -Steiner root of G . Nodes in $V(G)$ are called *real*, whereas nodes in $V(T) \setminus V(G)$ are called *Steiner*. We so define, for any $S \subseteq V(T)$: $\text{Real}(S) := S \cap V(G)$ and $\text{Steiner}(S) := S \setminus V(G)$. Two (sub)trees T, T' are *Steiner-equivalent*, denoted $T \equiv_G T'$, if and only if $\text{Real}(T) = \text{Real}(T') = S$ and there exists an isomorphism $\iota : V(T) \rightarrow V(T')$ such that $\iota(v) = v$ for any $v \in S$ (the trees are equal up to an appropriate identification of their Steiner nodes). Finally, given a node-subset $X \subseteq V(T)$, $T\langle X \rangle$ is the smallest subtree of T such that $X \subseteq V(T\langle X \rangle)$.

(Strongly) Chordal graphs. A *clique-tree* is a tree T_G whose nodes are the maximal cliques of G and such that for every $v \in V$, the maximal cliques containing v induce a subtree of T_G . A graph $G = (V, E)$ is called *chordal* if and only if it has a clique-tree. Moreover if G is chordal then, we can construct a clique-tree for G in $\mathcal{O}(m)$ -time [2]. An uv -separator is a subset $S \subseteq V \setminus \{u, v\}$ such that u and v are disconnected in $G \setminus S$. If in addition, no strict subset of S is an uv -separator then, S is a *minimal uv -separator*. A *minimal separator* of G is a minimal uv -separator for some $u, v \in V$. For a chordal graph G and *any* clique-tree T_G of G , S is a minimal separator of G if and only if there exist two maximal cliques K_i, K_j such that: $K_i K_j \in E(T_G)$ and $K_i \cap K_j = S$ [2]. We define $E_S(T_G) := \{K_i K_j \in E(T_G) \mid K_i \cap K_j = S\}$ (edges labeled by S).

A *clique-intersection* of G is the intersection of a subset of maximal cliques in G . The families of all clique-intersections, maximal cliques and minimal separators of G are denoted by $\mathcal{X}(G)$, $\mathcal{K}(G)$ and $\mathcal{S}(G)$, respectively. For a superclass of k -Steiner powers known as *strongly chordal graphs*, the family $\mathcal{X}(G)$ has polynomial size and can be computed in polynomial time [15].

Step 0 (Initialization). Given $G = (V, E)$, we check whether G is strongly chordal. If this is not the case then, G cannot be a 4-Steiner power, and we stop. Otherwise we compute $\mathcal{X}(G)$.

3 Structure Theorems

Some relationships between k -Steiner roots and clique-intersections are proved in Sec. 3.1. These structural results are the cornerstone of our algorithm and its analysis. Then, we refine our results for the special case $k = 4$ in Sec. 3.2.

3.1 Playing with the root

The following result is a generalization of [9, Lemma 1] to any k . We prove it by using some intricate properties of the eccentricity function on trees [14].

Theorem 2. *Given $G = (V, E)$ and T any k -Steiner root of G , the following properties hold for any clique-intersection $X \in \mathcal{X}(G)$:*

- We have $\text{Real}(T\langle X \rangle) = X$ and $\text{diam}(T\langle X \rangle) \leq k$;
- If $T' \supset T\langle X \rangle$ then, either $X = \text{Real}(T')$ or $\text{diam}(T') > \text{diam}(T\langle X \rangle)$;
- If $k = 2k'$ is even then, for any two different maximal cliques $K_i, K_j \in \mathcal{K}(G)$ we have $\mathcal{C}(T\langle K_i \rangle) \cap \mathcal{C}(T\langle K_j \rangle) = \emptyset$.

3.2 Well-structured 4-Steiner roots

For $k = 4$, we introduce new notions which only depend on the clique-intersections of G . Roughly, given $X \in \mathcal{X}(G)$ and an arbitrary root T , we introduce some operations in order to modify $T\langle X \rangle$. Doing so, we wish to force this subtree to have some more structure, thereby avoiding a combinatorial explosion of the number of possibilities to consider. Therefore, we carefully study the situations when a vertex $v \in X$ may not be arbitrarily movable inside $T\langle X \rangle$ (in which case we call v *X -constrained*). The most natural case is when there is a $X' \in \mathcal{X}(G)$ s.t. $X' \subset X$, $v \in X'$ and $|X'| \geq 2$ (v is *internally X -constrained*)⁴. However, more subtle cases occur when X is a minimal separator, or more generally X is contained in some larger clique-intersection. We say that v is *(X, X_1, X_2) -sandwiched* if $X_1, X_2 \in \mathcal{X}(G)$ are s.t. $X \subset X_1$ and $X \cap X_2 = \{v\} \subset X_1 \cap X_2$. Our study reveals that X -constrained vertices have a very rigid structure. Finally, a vertex that does not fall in one of these two above cases is called *X -free*. We prove that we can always force the X -free vertices to be *leaves* of the subtree $T\langle X \rangle$, thereby considerably reducing the number of possibilities for the latter.

Theorem 3. *Let $G = (V, E)$ be a 4-Steiner power. There always exists a well-structured 4-Steiner root T of G where, for any clique-intersection $X \in \mathcal{X}(G)$:*

⁴ When X is a *maximal clique*, the internally X -constrained vertices can be characterized in terms of simplicial vertices and a subset of the cut-vertices.

- all the X -free vertices are leaves of $T\langle X \rangle$ with maximum eccentricity $\text{diam}(T\langle X \rangle)$;
- there is a node $c \in \mathcal{C}(T\langle X \rangle)$ such that for every X -free vertex v , except maybe one, $\text{dist}_T(v, c) = \text{dist}_T(v, \mathcal{C}(T\langle X \rangle))$;
- all the internal nodes on a path between $\mathcal{C}(T\langle X \rangle)$ and a X -free vertex are Steiner nodes of degree two;
- and if $X \in \mathcal{K}(G)$ and it has a X -free vertex then, $\text{diam}(T\langle X \rangle) = 4$.

4 A special rooted clique-tree

We now present Step 1 of our algorithm so as to show all the steps in chronological order. However, please note that in the next Sec. 5, *any* clique-tree could be used. Indeed, we will only start using the peculiar properties of our rooted clique-tree in Sec. 6.

Step 1 (Construction of the rooted clique-tree). We construct a clique-tree T_G of G that we root in some $K_0 \in \mathcal{K}(G)$. In order to give the main intuition behind its construction, let us consider an arbitrary maximal clique K_i that is not the root, and its parent node $K_{p(i)}$. Let G_i be induced by the maximal cliques in the subtree of T_G rooted at K_i . If G has a 4-Steiner root then, by heredity, so does G_i . Roughly, we would like to bound the number of partial solutions for G_i that we will need to store for our dynamic programming algorithm. By Theorem 3, one first step for doing so would be to force most vertices in $S_i := K_i \cap K_{p(i)}$ to be K_i -free in the subgraph G_i . More specifically, for every descendant K_j of K_i in T_G we would like to impose $S_j \not\subseteq S_i$ and $S_i \not\subseteq S_j$ ⁵. However, both objectives are conflicting and so, we need to find a trade-off. Admittedly, our proposed solution is quite technical.

Given a clique-tree T_G of $G = (V, E)$, we say that a minimal separator S is *weakly T_G -convergent* if there exists some maximal clique K_S that is incident to all edges in $\bigcup_{S', S \subset S'} E_{S'}(T_G)$. S is termed *T_G -convergent* if it is weakly T_G -convergent and the maximal clique K_S is also incident to all edges in $E_S(T_G)$. The relationship between these notions and 4-Steiner roots is as follows:

Lemma 1. *Let T be any 4-Steiner root of G , and let $S \in \mathcal{S}(G)$. If $T\langle S \rangle$ is a non-edge star then, S is weakly T_G -convergent for any clique-tree T_G of G .*

Sketch proof. We may assume that S is strictly contained in some minimal separator S' . By Theorem 2, $T\langle S' \rangle$ has diameter three. This implies $\mathcal{C}(T\langle S \rangle) \subset \mathcal{C}(T\langle S' \rangle)$. Furthermore, we can prove that S' is contained in exactly two maximal cliques K_i, K_j and $\mathcal{C}(T\langle K_i \rangle) \cup \mathcal{C}(T\langle K_j \rangle) = \mathcal{C}(T\langle S' \rangle)$. Let us assume w.l.o.g. that $\mathcal{C}(T\langle S \rangle) = \mathcal{C}(T\langle K_i \rangle)$. Then, any minimal separator S'' that strictly contains S is contained in K_i and one other maximal clique $K_{S''}$. Let T_G be a clique-tree of G . We have $E_{S''}(T_G) \neq \emptyset$, and so $K_i K_{S''} \in E(T_G)$. By setting $K_S := K_i$, we get that S is weakly T_G -convergent. $\square$

⁵ Observe that if $S_j \subset S_i$, and $|S_j| \geq 2$, then the vertices of S_j are S_i -constrained.

Therefore, weak convergence is a *necessary* condition for a $S \in \mathcal{S}(G)$ to be contained in a star in some 4-Steiner root (that is the hardest case to deal with in our algorithm). If furthermore there is convergence then, we needn't store any costly information about the separators that strictly contain S in the encoding of partial solutions. Indeed, any “inclusion issue” between S and these separators can be handled with when we process the maximal clique K_S . So, we want to force weak convergence to imply convergence. Our construction in what follows applies to any S of size at least three. – For smaller separators, we can use much simpler counting arguments in order to bound the number of possible partial solutions that we will need to consider by a *constant*. See Sec. 6 for details. –

Theorem 4. *For any chordal graph G , we can compute in polynomial time a rooted clique-tree T_G where, for any $S_i := K_i \cap K_{p(i)}$:*

- *If S_i is weakly T_G -convergent and $|S_i| \geq 3$ then, S_i is T_G -convergent;*
- *Any minimal separator of G_i that is contained in S_i is T_G -convergent, has at least three vertices and is strictly contained in a minimal separator of G_i .*

5 A family of subtrees for the Clique-Intersections

Step 2 (Candidate set generation). We exploit a result of Section 3.1 which states that, for any 4-Steiner root T of G and for any clique-intersection X , the smallest subtree containing X does not contain any other real nodes. Then, our goal is, for every $X \in \mathcal{X}(G)$, to compute a polynomial-size family $\mathcal{T}_X$ of “candidate subtrees” whose real nodes are exactly X . Intuitively, $\mathcal{T}_X$ should contain all possibilities for $T(X)$ in a *well-structured* 4-Steiner root T (such a root must satisfy additional properties given in Sec. 3.2). Note that we only need to compute this above family for *minimal separators* and *maximal cliques*.

5.1 Case of Minimal Separators

The following result serves as a brick-basis construction for computing all the other families of candidate subtrees.

Theorem 5. *In $\mathcal{O}(n^5m)$ -time we can construct a collection $(\mathcal{T}_S)_{S \in \mathcal{S}(G)}$ such that, for any well-structured 4-Steiner root T of G , and for any $S \in \mathcal{S}(G)$, $T(S)$ is Steiner-equivalent to some subtree in $\mathcal{T}_S$.*

Sketch proof. Let us describe the main difficulty we had to face on in order to prove this above result. Given $S \in \mathcal{S}(G)$ the difficulty in generating $\mathcal{T}_S$ comes from the bistars (diameter-three subtrees), as a brute-force generation of all possibilities would take time exponential in $|S|$. Let $\mathcal{X}(S) = \{X \in \mathcal{X}(G) \mid X \subset S, |X| \geq 2\}$. Based on a careful analysis of the intersection graph $I_S = (\mathcal{X}(S), \{XX' \mid X \cap X' \neq \emptyset\})$, we can bound the number of possible mappings of the internally S -constrained vertices to the nodes of a bistar by an $\mathcal{O}(|S|^2)$. We can also bipartition the sandwiched vertices in such a way that each group

should be mapped to a different side of the bistar; each group should in fact correspond to one of the two maximal cliques containing S . Then, we use the fact that in a well-structured 4-Steiner root of G , S -free vertices are leaves of such a bistar with all of them, except maybe one, adjacent to the same central node. For a fixed mapping of the S -constrained vertices, this only gives us $\mathcal{O}(|S|)$ possibilities in order to map the S -free vertices. Overall, we reduce the number of possible bisters to an $\mathcal{O}(|S|^5)$. $\square$

5.2 Case of a Leaf Node

Theorem 6. *Given $G = (V, E)$ and a rooted clique-tree T_G of G , let $K_i \in \mathcal{K}(G)$ be a leaf. We can construct, in time polynomial in $|K_i|$, a set $\mathcal{T}_i$ of 4-Steiner roots for $G_i := G[K_i]$ with the following additional property: In any well-structured 4-Steiner root T of G , there exists a $T'_i \in \mathcal{T}_i$ Steiner-equivalent to $T\langle K_i \rangle$.*

Sketch proof. We use a well-known decomposition of K_i into a unique minimal separator $S_i := K_i \cap K_{p(i)}$ and a set of simplicial vertices. Given any fixed possibility for $T\langle S_i \rangle$, there are $\mathcal{O}(|S_i|)$ possibilities for $T\langle S_i \cup \mathcal{C}(T\langle K_i \rangle) \rangle$. Then, we use the fact that all simplicial vertices are K_i -free. Since we already fixed $\mathcal{C}(T\langle K_i \rangle)$, by Theorem 3, there is essentially one way to add the K_i -free vertices in order to complete the construction (up to Steiner equivalence). $\square$

5.3 Case of an internal node

Finally, we consider the maximal cliques K_i that are internal nodes of T_G . Unsurprisingly, several new difficulties arise in the construction of $\mathcal{T}_{K_i}$. Our bottleneck is solving the following subproblem: compute (up to Steiner equivalence) all possible central nodes and their neighbourhood in any subtree $T\langle K_i \rangle$ of diameter four. We solved this subproblem in most situations, *e.g.*, when there is a minimal separator $S \subseteq K_i$ such that $T\langle S \rangle$ must be a bistar (diameter-three subtree). For that, we combine some key arguments in the proof of Theorem 5 with the transformation techniques that we used in the proof of Theorem 3.

Lemma 2. *For any graph G , let $S \in \mathcal{S}(G)$, let K be a maximal clique containing S and let R, c be such that $R \subset S$ and either $c \in R$ or c is Steiner. We can compute in $\mathcal{O}(nm \log n)$ -time a node c' with the following properties: For any well-structured 4-Steiner root T of G s.t. $T\langle S \rangle$ is a bistar, $c \in \mathcal{C}(T\langle S \rangle) \setminus \mathcal{C}(T\langle K \rangle)$, and $\text{Real}(N_T[c]) = R$, there exists a well-structured root T' with the same properties s.t. $\mathcal{C}(T'\langle K \rangle) = \{c'\}$, and $\text{dist}_{T'}(u, v) \geq \text{dist}_T(u, v)$ for every $u, v \in V$; moreover, either $T \equiv_G T'$, or $\sum_{u, v \in V} \text{dist}_{T'}(u, v) > \sum_{u, v \in V} \text{dist}_T(u, v)$.*

In order to better understand the significance of Lemma 2, assume that $T\langle S \rangle$ should be a bistar in the final solution we want to compute, and that we already identified one of its center node c and the set of real nodes R to which c must be adjacent. What this above property says is that there is essentially one canonical way to compute the bistar given R and c . The more technical condition $\text{dist}_{T'}(u, v) \geq \text{dist}_T(u, v)$ is simply there in order to ensure that by doing

so, we cannot miss a solution of an intermediate problem we call DISTANCE-CONSTRAINED ROOT (*i.e.*, see Sec. 6). Finally, our condition on the potential function $\sum_{u,v \in V} \text{dist}_{T'}(u,v)$ increasing ensures that we can repeatedly apply our “canonical completion” method for arbitrarily many minimal separators S . By using this above method, we obtain the following intermediate construction:

Lemma 3. *For any chordal graph G , let T_G be a rooted clique-tree and let K_i be a maximal clique of $G = (V, E)$ with no K_i -free vertex. In $\mathcal{O}(|K_i|^6 \cdot n^3 m \log n)$ -time, we can compute a family $\mathcal{B}_i$ with the following special property: For any well-structured 4-Steiner root T of G where for at least one minimal separator $S \subset K_i$, $T\langle S \rangle$ is a bistar, there is a T' such that $T'\langle S_i \rangle \equiv_G T\langle S_i \rangle$, $T'\langle K_i \rangle \in \mathcal{B}_i$ and $\text{dist}_{T'}(r, V(G_i) \setminus S_i) \geq \text{dist}_T(r, V(G_i) \setminus S_i)$ for every $r \in V(T\langle S_i \rangle)$ ⁶.*

Note that we do not capture *all* well-structured roots with this above lemma, but only those maximizing certain distances’ conditions. In the remaining cases when there are no minimal separators S that are mapped to a bistar, our construction is less satisfying. Specifically, we are left with some “problematic subsets” called *thin branches*: with exponentially many possible ways to include them in candidate subtrees. As a way to circumvent this combinatorial explosion, we also include in $\mathcal{T}_{K_i}$ some partially constructed subtrees where the thin branches are omitted. We will greedily decide how to include the thin branches in these subtrees at Step 4 (Sec. 7).

6 Deciding the partial solutions to store

Step 3 (Selection of the encodings). For the remainder of the algorithm, let $(K_q, K_{q-1}, \dots, K_0)$ be a post-ordering of the maximal cliques (*i.e.*, obtained by depth-first-search traversal of our rooted clique-tree T_G). We consider the maximal cliques $K_i \in \mathcal{K}(G)$ sequentially, from $i = q$ down to $i = 0$. The next two Sections are devoted to the computation of a subset $\mathcal{T}_i$ of 4-Steiner roots for G_i . Specifically, for any 4-Steiner root T_i of G_i we define the following encoding:

$$\text{encode}(T_i) := \left[T_i\langle S_i \rangle \mid (\text{dist}_{T_i}(r, V(G_i) \setminus S_i))_{r \in V(T_i\langle S_i \rangle)} \right].$$

In what follows, we compute a polynomial-size subset of *allowed* encodings for the partial solutions in $\mathcal{T}_i$. That is, we only want to add in $\mathcal{T}_i$ some partial solutions for which the encoding is in the list. Formally, we define an auxiliary problem called DISTANCE-CONSTRAINED ROOT, where given an encoding as input, we ask whether there exists a corresponding 4-Steiner root of G_i .

⁶ Recall that S_i and G_i were defined in Sec. 4. By convention, $S_i = \emptyset$ if K_i is the root.

Problem 2 (DISTANCE-CONSTRAINED ROOT).

Input: a graph $G = (V, E)$ with a rooted clique-tree T_G , a maximal clique K_{i_j} , a tree $T_{S_{i_j}}$ s.t. $\text{Real}(T_{S_{i_j}}) = S_{i_j}$, and a sequence $(d_r)_{r \in V(T_{S_{i_j}})}$ of positive integers.

Output: Either a 4-Steiner root T_{i_j} of G_{i_j} s.t. $T_{S_{i_j}} \equiv_G T_{i_j} \langle S_{i_j} \rangle$ and, $\forall r \in V(T_{S_{i_j}})$: $\text{dist}_{T_{i_j}}(r, V(G_{i_j}) \setminus S_{i_j}) \geq d_r$; Or $\perp$ if there is no such a T_{i_j} which can be extended to some *well-structured* 4-Steiner root T of G .

Theorem 7. *Given $G = (V, E)$ chordal and a rooted clique-tree T_G as in Theorem 4, let K_i be an internal node with children $K_{i_1}, K_{i_2}, \dots, K_{i_p}$. If we can solve DISTANCE-CONSTRAINED ROOT in time $P(n, |S_{i_j}|)$ for some polynomial P then, we can compute in time $\mathcal{O}(n|K_i|^5 P(n, |K_i|))$ a family $\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \dots, \mathcal{T}_{i_p}$ of 4-Steiner roots for $G_{i_1}, G_{i_2}, \dots, G_{i_p}$, respectively, such that:*

1. *For any $j \in \{1, 2, \dots, p\}$, $|\mathcal{T}_{i_j}| = \mathcal{O}(|S_{i_j}|^5)$;*
2. *For any well-structured 4-Steiner root T of G , there exists a T' such that: $T \langle K_i \rangle \equiv_G T' \langle K_i \rangle$, $T' \langle V(G_{i_j}) \rangle \in \mathcal{T}_{i_j}$ for any $j \in \{1, 2, \dots, p\}$, and $\text{dist}_{T'}(r, V(G_i) \setminus S_i) \geq \text{dist}_T(r, V(G_i) \setminus S_i)$ for any node $r \in V(T \langle S_i \rangle)$.*

Sketch proof. We process the children nodes K_{i_j} sequentially by non-decreasing size of the minimal separators S_{i_j} . For that, we start constructing the family $\mathcal{T}_{S_{i_j}}$ of Theorem 5, and we consider the subtrees $T_{S_{i_j}} \in \mathcal{T}_{S_{i_j}}$ sequentially. We divide the proof into several cases depending on $|S_{i_j}|$ and on $\text{diam}(T_{S_{i_j}})$.

Case $|S_{i_j}| \leq 2$. There can only be $\mathcal{O}(1)$ different possibilities for the distances $(d_r)_{r \in V(T_{S_{i_j}})}$. We could solve DISTANCE-CONSTRAINED ROOT for all these possibilities, thereby obtaining the family $\mathcal{T}_{i_j}$. But in fact, this seemingly simple case hides a time bomb that will detonate during the second part of the proof (*i.e.*, when we consider larger minimal separators). To understand why through an example, let us assume the existence of a large separator S_{i_k} of which every vertex is also a cut-vertex. Then, one possibility for $T_{S_{i_k}}$ is a star with a Steiner central node. For every leaf-node v of that star, let us consider a maximal clique K_{i_j} s.t. $S_{i_j} = \{v\}$. The star $T_{S_{i_k}}$ can only be compatible with solutions T_{i_j} s.t. $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus \{v\}) \geq 5 - \text{dist}_{T_{S_{i_k}}}(v, S_{i_k} \setminus \{v\}) = 3$. In particular, we may have up to *two* compatible solutions T_{i_j} , and that gives us in turn two different possibilities for the constraint $\text{dist}_{T_{i_k}}(v, V(G_{i_k}) \setminus S_{i_k})$. But then, since this is true for any $v \in S_{i_k}$, we are left with $2^{|S_{i_k}|}$ possibilities for the distance constraints $(d_r)_{r \in V(T_{S_{i_k}})}$! We can resolve this issue by always choosing any compatible solution which *maximizes* $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus \{v\})$. Specifically, if $S_{i_j} = \{v\}$ is a cut-vertex then, we only keep in the family $\mathcal{T}_{i_j}$ the partial solution maximizing $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j})$. In the same way, if $S_{i_j} = \{u, v\}$ and $T_{S_{i_j}}$ is an edge then, we only need to keep two solutions, namely: among all those maximizing $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j})$ (resp., $\text{dist}_{T_{i_j}}(u, V(G_{i_j}) \setminus S_{i_j})$) the one maximizing $\text{dist}_{T_{i_j}}(u, V(G_{i_j}) \setminus S_{i_j})$ (resp., $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j})$).

Case $|S_{i_j}| \geq 3$. The processing of large minimal separators S_{i_j} is more intricate. For a fixed $T_{S_{i_j}}$ we define a family of shorter encodings with only $|S_{i_j}|^{\mathcal{O}(1)}$ possibilities, that essentially summarizes at “guessing” the central nodes of $T\langle K_i \rangle$ and $T\langle K_{i_j} \rangle$. Assuming a correct guess of these above central nodes, for any partial solution T_{i_j} that is compatible with $T_{S_{i_j}}$, we show how to extract a *constant-number* of distance constraints from $\text{encode}(T_{i_j})$, in such a way that all other constraints can be retrieved from those $\mathcal{O}(1)$ that we keep in the short encoding or proved to be irrelevant. Overall, we show that it is sufficient to store only one solution per possible short encoding. For the purpose of illustration, let us focus on the case when $T_{S_{i_j}}$ is a star (the case of bistars is similar, but simpler). We first assume that no minimal separator of G_{i_j} contains S_{i_j} . We may further assume that no minimal separator of G_{i_j} can be contained in S_{i_j} (otherwise, by the second property of Theorem 4 such separators should have size at least 3, whereas since $T_{S_{i_j}}$ is a star they should have size at most 2). In our first subcase, we assume that the center c of the star will *not* end in $\mathcal{C}(T_{i_j}\langle K_{i_j} \rangle)$. Then, we prove that for every leaf v of the star except maybe one, $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j}) = \text{dist}_{T_{i_j}}(c, V(G_{i_j}) \setminus S_{i_j}) + 1$ (two distances to store in the short encoding). Otherwise, $c \in \mathcal{C}(T_{i_j}\langle K_{i_j} \rangle)$. Our previous formula for $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j})$ stays true unless v is also contained in a minimal separator of G_{i_j} . In this latter case, such a minimal separator must *overlap* S_{i_j} , and so we can prove that we always have $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j}) = 1$. Finally, we assume that a minimal separator of G_{i_j} contains S_{i_j} . We derive from both properties of Theorem 4 that $c \in \mathcal{C}(T_{i_j}\langle K_{i_j} \rangle)$ and S_{i_j} is T_G -convergent, with $K_{S_{i_j}} = K_{i_j}$. In particular, $S_{i_j} \not\subseteq S_i$, and so, $|S_{i_j} \cap S_i| \leq 2$. Recall that we started by guessing $\mathcal{C}(T\langle K_i \rangle)$ and $\mathcal{C}(T\langle K_{i_j} \rangle)$. We include in our short encoding from the previous case the distances $\text{dist}_{T_{i_j}}(v, V(G_{i_j}) \setminus S_{i_j})$, $v \in S_{i_j} \cap (S_i \cup \mathcal{C}(T\langle K_i \rangle))$, plus some *fixed* additional constraints that are derived from the smaller separators contained in S_{i_j} . We stress that this approach could not work with an *arbitrary* T_G . $\square$

7 The dynamic programming

Step 4 (Greedy strategy). While we execute Step 3 for its father node $K_{p(i)}$, we compute for K_i a polynomial-size subset of allowed encodings for the 4-Steiner roots of G_i which we want to compute. For all the constraints in such encodings, we are left to decide whether there exists a 4-Steiner root of G_i which satisfies all of them (*i.e.*, we must solve DISTANCE-CONSTRAINED ROOT).

Theorem 8. *For every strongly chordal graph G , let $\|G\| := \sum_{K_i \in \mathcal{K}(G)} |K_i|$. Let T_G be a rooted clique-tree as in Theorem 4 and let $K_i \in \mathcal{K}(G)$. There is some polynomial P such that, after a pre-processing in time $\mathcal{O}(n\|G_i\|^5 P(n))$, we can solve DISTANCE-CONSTRAINED ROOT for any input $T_{S_i}, (d_r)_{r \in V(T_{S_i})}$ in time $\mathcal{O}(P(n))$.*

Sketch proof. If K_i is a leaf of T_G then, we construct the family given by Theorem 6. We keep the trees $T_i \in \mathcal{T}_i$ that satisfy the constraints we have. From

now on, let us assume K_i is internal with children $K_{i_1}, K_{i_2}, \dots, K_{i_p}$. We start by computing $\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \dots, \mathcal{T}_{i_p}$ as in Theorem 7. We also need to construct a representation of the family $\mathcal{T}_{K_i}$, as sketched in Sec. 5.3. Roughly, the elements in this representation are of the form $(T_{Y_i}, \mathcal{C}_i)$ where $Y_i \subseteq K_i$ and $\mathcal{C}_i$ must represent the center of $T\langle K_i \rangle$ (missing vertices of $K_i \setminus Y_i$ are supposed to be located in thin branches). This ends the pre-processing step for K_i . In what follows let T_{S_i} and $(d_r)_{r \in T_{S_i}}$ be fixed. In order to solve DISTANCE-CONSTRAINED ROOT, we start by enumerating all pairs $(T_{Y_i}, \mathcal{C}_i)$. Our construction ensures that $S_i \subseteq Y_i$ and so, we can check whether $T_{S_i} \equiv_G T_{Y_i}\langle S_i \rangle$. If this not the case then, we can withdraw this pair and continue. Due to lack of space, we now only sketch the case $Y_i = K_i$ (no thin branch). For every $r \in T_{S_i}$ we check whether we have: $\text{dist}_{T_{Y_i}}(r, K_i \setminus S_i) \geq d_r$ (otherwise, we violate our distances' constraints). In the same way, for every $j \in \{1, \dots, p\}$, we remove from $\mathcal{T}_{i_j}$ any partial solution T_{i_j} s.t. either $T_{i_j}\langle S_{i_j} \rangle \neq T_{Y_i}\langle S_{i_j} \rangle$ or the distances' constraints are violated. We finally explain how to greedily construct a solution (if any), starting from $T_i := T_{Y_i}$. The procedure is divided into a constant number of phases. Every time we complete one of these phases, we select a $T_{i_j} \in \mathcal{T}_{i_j}$, for some j , then we remove from all other $\mathcal{T}_{i_k}$'s the incompatible partial solutions.

Phase 1: Processing the cut-vertices. We consider all the indices j s.t. $S_{i_j} = \{v\}$ is a cut-vertex. There is *one* solution left in $\mathcal{T}_{i_j}$, so we need to add it.

Phase 2: Processing the edges. We consider all the indices j s.t. $S_{i_j} = \{u, v\}$ and $T_{Y_i}\langle S_{i_j} \rangle$ is an edge. We show that we can almost proceed similarly as for Phase 1 provided we know which among u or v will be closest to $V(G_i) \setminus V(G_{i_j})$ in a final solution. Therefore, computing this information is the main objective of this phase. In general, we will pick a vertex of S_{i_j} which is the closest to $\mathcal{C}_i$, but several cases need to be considered before we can validate such a choice.

Phase 3: Processing the bistars. We consider all the indices j s.t. $T_{Y_i}\langle S_{i_j} \rangle$ is a bistar. A careful analysis shows that in all cases but one degenerate, we can select any $T_{i_j} \in \mathcal{T}_{i_j}$ s.t. $\text{dist}_{T_{i_j}}(\mathcal{C}_i, V(G_{i_j}) \setminus S_{i_j})$ is maximized.

Phase 4: Processing the stars. We finally consider all the indices j s.t. $T_{Y_i}\langle S_{i_j} \rangle$ is a star. Let $\mathcal{C}(T_{Y_i}\langle S_{i_j} \rangle) = \{c\}$. Due to lack of space, we only describe the subcase $c \in \mathcal{C}_i$, which is simpler⁷. We first prove that for every unprocessed $S_{i_k} \neq S_{i_j}$, a best possible choice would be to pick a $T_{i_j} \in \mathcal{T}_{i_j}$ s.t. $\text{dist}_{T_{i_j}}(\mathcal{C}(T_{i_j}\langle K_{i_j} \rangle), V(G_{i_j}) \setminus S_{i_j})$ is maximized. However, we also need to account for the other indices k such that $S_{i_k} = S_{i_j}$. For that, let $J = \{j' \mid S_{i_{j'}} = S_{i_j}\}$. The solutions $T_{i_{j'}}, j' \in J$ that we will choose must have diameter four, and the center nodes $v_{j'}$ in $T_{i_{j'}}\langle K_{i_{j'}} \rangle$ must be pairwise different. We do a reduction to MAXIMUM-WEIGHT MATCHING where we create a bipartite graph with respective partite sets J and all possible central nodes. For every $j' \in J$ and $T_{i_{j'}} \in \mathcal{T}_{i_{j'}}$ of diameter four, we add an edge $\{j', \mathcal{C}(T_{i_{j'}})\}$ of weight $\text{dist}_{T_{i_{j'}}}(\mathcal{C}(T_{i_j}\langle K_{i_{j'}} \rangle), V(G_{i_{j'}}) \setminus S_{i_{j'}})$. $\square$

⁷ All the missing cases, that includes the addition of thin branches to T_{Y_i} , are solved by using the same matching-based approach as in this subcase.

References

1. S. Arumugam, A. Brandstädt, T. Nishizeki, and K. Thulasiraman. *Handbook of graph theory, combinatorial optimization, and algorithms*. 2016.
2. J. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. 1993.
3. J. A. Bondy and U. S. R. Murty. *Graph theory*. 2008.
4. A. Brandstädt and V. Le. Structure and linear time recognition of 3-leaf powers. *Information Processing Letters*, 98(4):133–138, 2006.
5. A. Brandstädt, V. Le, and D. Rautenbach. Exact leaf powers. *Theoretical Computer Science*, 411(31-33):2968–2977, 2010.
6. A. Brandstädt, V. Le, and R. Sritharan. Structure and linear-time recognition of 4-leaf powers. *ACM Transactions on Algorithms (TALG)*, 5(1):11, 2008.
7. A. Brandstädt and P. Wagner. Characterising (k, ℓ) -leaf powers. *Discrete Applied Mathematics*, 158(2):110–122, 2010.
8. T. Calamoneri and B. Sinaimeri. Pairwise compatibility graphs: A survey. *SIAM Review*, 58(3):445–460, 2016.
9. M. Chang and M. Ko. The 3-Steiner root problem. In *WG*, pages 109–120, 2007.
10. G. Ducoffe. Polynomial-time Recognition of 4-Steiner Powers. Technical Report arXiv:1810.02304, ArXiv, 2018.
11. D. Eppstein and E. Havvaei. Parameterized Leaf Power Recognition via Embedding into Graph Products. In *IPEC’18*, pages 16:1–16:14, 2019.
12. M. Fellows, D. Meister, F. Rosamond, R. Sritharan, and J. Telle. Leaf powers and their properties: Using the trees. In *ISAAC*, pages 402–413, 2008.
13. T. Jiang, P. Kearney, and G. Lin. Phylogenetic k-root and steiner k-root. In *ISAAC*, pages 539–551, 2000.
14. C. Jordan. Sur les assemblages de lignes. *J. Reine Angew. Math*, 70(185):81, 1869.
15. R. Nevries and C. Rosenke. Characterizing and computing the structure of clique intersections in strongly chordal graphs. *Discr. Applied Math.*, 181:221–234, 2015.
16. R. Nevries and C. Rosenke. Towards a characterization of leaf powers by clique arrangements. *Graphs and Combinatorics*, 32(5):2053–2077, 2016.
17. N. Nishimura, P. Ragde, and D. Thilikos. On graph powers for leaf-labeled trees. *Journal of Algorithms*, 42(1):69–108, 2002.