On the Complexity of Parameterized Local Search for the Maximum Parsimony Problem

Christian Komusiewicz
Institute of Computer Science, Friedrich Schiller Universität Jena, Germany

Simone Linz
School of Computer Science, University of Auckland, New Zealand

Nils Morawietz
Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, Germany

Jannik Schestag
Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, Germany

Abstract

Maximum Parsimony is the problem of computing a most parsimonious phylogenetic tree for a taxa set X from character data for X. A common strategy to attack this notoriously hard problem is to perform a local search over the phylogenetic tree space. Here, one is given a phylogenetic tree T and wants to find a more parsimonious tree in the neighborhood of T. We study the complexity of this problem when the neighborhood contains all trees within distance k for several classic distance functions. For the nearest neighbor interchange (NNI), subtree prune and regraft (SPR), tree bisection and reconnection (TBR), and edge contraction and refinement (ECR) distances, we show that, under the exponential time hypothesis, there are no algorithms with running time $|I|^{o(k)}$ where $|I|$ is the total input size. Hence, brute-force algorithms with running time $|X|^{\mathcal{O}(k)} \cdot |I|$ are essentially optimal.

In contrast to the above distances, we observe that for the sECR-distance, where the contracted edges are constrained to form a subtree, a better solution within distance k can be found in $k^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$ time.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Applied computing → Molecular evolution; Applied computing → Computational genomics

Keywords and phrases phylogenetic trees, parameterized complexity, tree distances, NNI, TBR

Digital Object Identifier 10.4230/LIPIcs.CPM.2023.18

Funding CK and SL were supported by Catalyst Seeding funding provided by the New Zealand Ministry of Business, Innovation and Employment and administered by the Royal Society of New Zealand Te Apārangi. NM was supported by the Deutsche Forschungsgemeinschaft (DFG), project OPERAH, KO 3669/5-1.

1 Introduction

Maximum Parsimony is one of the most popular methods for inferring phylogenetic (evolutionary) trees from sequences of morphological or molecular characters. Given sequences of characters for n taxa, this method reconstructs a phylogenetic tree T whose n leaves are labeled bijectively by the n taxa and that has the minimum *parsimony score* over all such trees. The parsimony score is the number of character state changes along the tree edges that are necessary when extending the sequences for the leaves of T to all internal vertices of T. Note that for each character, this score is at least $s - 1$, where s denotes the number

34th Annual Symposium on Combinatorial Pattern Matching (CPM 2023).
Editors: Laurent Bulteau and Zsuzsanna Lipták; Article No. 18; pp. 18:1–18:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of different character states. A phylogenetic tree is called perfect if it achieves score $s-1$ for every character. Such a perfect phylogeny does not always exists. For a more comprehensive introduction to Maximum Parsimony, we refer the interested reader to [9].

From an algorithmic point of view, the Maximum Parsimony problem is notoriously hard: It is NP-complete even for binary characters [12]. Moreover, the current best running time is $\Omega((2n-3)!!)$, where $(2n-3)!! = 1 \cdot 3 \cdot \ldots \cdot (2n-5) \cdot (2n-3)$ [4]. The associated algorithm generates all possible binary phylogenetic trees on n leaves in a bottom-up fashion. Hence, the best known algorithm is essentially a brute-force-method. This running time bound is impractical when $n > 15$. Better running times are possible when the instance has a near-perfect phylogeny and the number of different character states s is small. Here, the running time is measured also in terms of the excess q over the score of a perfect phylogeny. In the general case, Maximum Parsimony can be solved in $nm^{\mathcal{O}(q)}2^{\mathcal{O}(q^2 s^2)}$ time [10], where m is the length of the character sequences. In 2007, the running time was improved to $\mathcal{O}(21^q + 8^q nm^2)$ for the special case of binary characters and the practical usefulness of the improved algorithm was demonstrated for $q \leq 10$ [30]. In the worst case, however, q can be essentially as large as m. Moreover, Maximum Parsimony is NP-hard even for $q = 0$ when the number of different character states is unbounded [3].

Given the hardness of Maximum Parsimony, solving this problem exactly is impractical for many real-world datasets due to prohibitive running times. Consequently, heuristic approaches, in particular local search, play an important role in computing good, but not necessarily optimal, solutions [2, 13, 14, 16, 17, 18, 19, 25, 26]. These approaches search the space of all possible phylogenetic trees on n taxa. In the course of such a search, the parsimony score of a subset of the phylogenetic trees in the space is computed. For any given tree, this step takes polynomial time using Fitch's or Sankoff's algorithm [11, 28]. A search through tree space starts by first computing a starting tree T before computing the parsimony score of all neighbors of T. If there is a neighboring tree T' whose parsimony score is smaller than that of T, then the search is continued by computing the parsimony score of all neighbors of T' and so on until a local optimum is found. In each iteration of the search, the neighboring trees are those that can be obtained from the current best tree by one or more rearrangement operations. The most well-known rearrangement operations on trees that are also considered in local search approaches for Maximum Parsimony, are nearest neighbor interchange (NNI), subtree prune and regraft (SPR), and tree bisection and reconnection (TBR) [1]. Each of these operations deletes an edge of a tree and then reconnects the resulting two subtrees. Depending on the operation, the reconnection is more or less restrictive, with SPR being a generalization of NNI and TBR being a generalization of SPR. The set of all trees that can be obtained by one operation is called the NNI, SPR, or TBR neighborhood, respectively. More general, we say that a tree T' is in the k-neighborhood with respect to NNI, SPR, or TBR of another tree T, if T' can be obtained from T by at most k NNI, SPR, or TBR operations, respectively.

In addition to NNI, SPR, and TBR, the k-ECR operation has also been considered in the literature (see for example the works by Ganapathy et al. [13, 14]). This latter operation first contracts up to k edges and then refines the resulting tree arbitrarily. Here, the k-ECR neighborhood contains all trees that can be obtained from a starting tree by applying one k-ECR operation. The 1-ECR neighborhood is exactly the NNI neighborhood, but the 2-ECR neighborhood strictly contains the set of trees reachable by two NNI moves [14]. The k-ECR neighborhood appeared earlier implicitly under the term *sectorial search* [19]. The k-sECR neighborhood, a restricted version of the k-ECR neighborhood where the contracted edges must form a subtree was considered by Sankoff et al. [29]. They found that for larger values of k, the k-sECR neighborhood gives better results than the 1-ECR neighborhood or, equivalently, the NNI neighborhood. Guo et al. [20] found that exploring the k-ECR neighborhood is too costly and thus proposed a restriction of this neighborhood which already leads to very good local optima. Their approach contracts k edges and then refines the resulting tree by using neighbor joining, a fast distance-based method to reconstruct phylogenetic trees. To summarize, local search is an important paradigm for designing heuristics for Maximum Parsimony, and it has been noted that larger neighborhoods such as the k-ECR neighborhood give better results at the cost of higher running times. So far, there is however no study of how hard exploring larger neighborhoods actually is.

To analyze the computational complexity of exploring neighborhoods under NNI, SPR, TBR, k-ECR, and k-sECR, we use the framework of *parameterized local search* [8, 15, 23, 24]. Here, one studies local search problems with a neighborhood whose size can be adjusted by a parameter k. In the canonical parameterized local search problem, one is then given some solution for an optimization problem and the question is whether there is a better solution in the k-neighborhood. Local search for any of the aforementioned neighborhoods that are associated with distances between two trees fits exactly into this framework: we are given a phylogenetic tree and want to know whether there is one with a better parsimony score in the k-neighborhood. Typically, the k-neighborhood has a size of $\mathcal{O}(|I|^{f(k)})$, where $|I|$ is the input size. In our case, the input size $|I|$ is in $\mathcal{O}(n^2 \cdot m)$. Thus, using a brute-force algorithm, one can find a better solution in the neighborhood if it exists in $|I|^{f(k)}$ time. The algorithmic question is now whether this can be done much faster. In particular, a running time of $f(k) \cdot |I|^{\mathcal{O}(1)}$ would be desirable since the explosion in the running time would then depend only on k and not on $|I|$. Parameterized algorithmics provides toolkits to design such algorithms or to show that such algorithms are unlikely. The latter can be done by showing W[1]-hardness with respect to k [6, 7] or by giving tight running time bounds based on the exponential time hypothesis (ETH) [21].

Our results are as follows. We show that even when all characters are binary, searching the k-ECR neighborhood is W[1]-hard with respect to k. The reduction that we use to establish this result also shows that, under the ETH, a running time of $|I|^{\Omega(k)}$ is necessary. Moreover, the reduction implies hardness for searching the k-neighborhood with respect to NNI, SPR, and TBR. In a nutshell, our results show that one cannot gain a substantial speed-up over the brute-force algorithm when trying to search these large neighborhoods. We then establish that $n^{\mathcal{O}(k)} \cdot m$ time is sufficient to search the k-neighborhoods with respect to any of NNI, SPR, TBR, and k-ECR, giving tight upper and lower bounds for the running time dependence on k. Finally, we observe that the k-sECR neighborhood of Sankoff [29] can be searched in $k^{\mathcal{O}(k)} \cdot |I|^{\mathcal{O}(1)}$ time, making it possible to consider much larger values of k than for the other neighborhoods. Let us remark that, while we formally study the decision problem that asks for the existence of a better tree in the k-neighborhood, our hardness results and algorithms also apply to the problem of finding an optimal tree in the k-neighborhood.

Proofs of statements marked with (*) are deferred to a full version of the article.

2 Preliminaries

For details about relevant definitions of parameterized complexity such as fixed-parameter tractability, W[1]-hardness, parameterized reductions and ETH, refer to the standard monographs [6, 7].

Graph notation. For a graph $G = (V, E)$ and a vertex set $K \subseteq V$, let $E(K)$ denote the set of edges of G where both endpoints are from K. The *subdivision of an edge* $e \in E$ *in* G results in the graph G' obtained by removing e from G and adding a new vertex which is adjacent to both endpoints of e. Let v be a vertex of degree 2 in G. The *suppression of* v in G results in the graph G' obtained by removing v from G and joining both neighbors of v by an edge.

Phylogenetic trees. Throughout this paper, X denotes a non-empty finite set of *taxa*.

An *unrooted phylogenetic* X*-tree* (for short, X*-tree*) T is a tree with leaf-set X and where no vertex has degree 2. If all non-leaf vertices of T have degree three, then T is called *binary*. Furthermore, if an edge e is incident with a leaf of T, then e is called a *pendant edge* and, otherwise, an *internal edge*. For two disjoint sets of taxa A and B, we say that $A|B$ is a *split* of an X-tree T if there is an edge e in T such that the deletion of e results in two subtrees where one has leaf set A and the other has leaf set B. The set of all splits of T is denoted by $\Sigma(T)$. Furthermore, we say that an X-tree T' is a *refinement* of T if $\Sigma(T) \subseteq \Sigma(T')$. Additionally, if T' is binary, then T' is a *binary refinement* of T. We say that two X-trees T and T' are *isomorphic* if $\Sigma(T) = \Sigma(T')$. Equivalently, two X-trees T and T' are *isomorphic* if there is a bijection φ between the vertices of T and the vertices of T' such that $\varphi(x) = x$ for all $x \in X$, and for all distinct vertices u and v of T, $\{u, v\}$ is an edge of T if and only if $\{\varphi(u), \varphi(v)\}$ is an edge of T'.

Now, let T be an X-tree and let V' be a subset of the vertices of T. Then $T(V')$ denotes the minimal subtree of T containing all vertices in V'. Let A be a non-empty and proper subset of X and let T be a binary X-tree. If $A|(X \setminus A)$ is a split of T, then the subtree $T(A)$ is a *pendant* A*-tree*. Moreover, the *pseudo-root* of $T(A)$ is the unique vertex of degree 2 in $T(A)$ if $|A| > 1$ and the unique vertex of $T(A)$, otherwise.

Maximum parsimony. A *character*[1] c on X is a function $c : X \to C$. If $|C| = 2$, then c is called a *binary* character. Intuitively, C can be thought of as the underlying alphabet and each element in the alphabet is a *character state*. Let T be an X-tree with vertex set V, and let c be a character on X whose set of character states is C. An *extension* c^* of c to V is a function $c^* : V \to C$ such that $c^*(x) = c(x)$ for each taxon $x \in X$. Let c^* be an extension of c. A *mutation edge of* c^* *in* T is an edge $\{u, v\}$ in T such that $c^*(u) \neq c^*(v)$ and we let $\text{score}_{c^*}(T)$ denote the number of mutation edges of c^* in T. Then the *parsimony score* of c on T, denoted by $\text{score}_c(T)$, is obtained by minimizing $\text{score}_{c^*}(T)$ over all possible extensions c^* of c. An extension c^* that minimizes $\text{score}_{c^*}(T)$ is called an *optimal extension of* c *in* T. Moreover the *maximum parsimony score* of c, denoted by $\text{MP}(c)$, is the parsimony score of c minimized over all binary X-trees.

Now let $S = (c_1, c_2, \ldots, c_m)$ be a sequence of characters on X. Then the parsimony score of S on an X-tree T is defined as $\text{score}_S(T) = \sum_{i=1}^{m} \text{score}_{c_i}(T)$ and, similarly, the maximum parsimony score of S, denoted by $\text{MP}(S)$, is the parsimony score of S minimized over all binary X-trees.

We may abuse notation by writing $c \in S$ if the character c is contained in the sequence S.

SPR and TBR. Let T be a binary X-tree. Let $e = \{u, v\}$ be an edge of T, and let T_1 and T_2 be the two trees obtained from T by deleting e and suppressing u if its degree is 2. Without loss of generality, we may assume that T_2 contains v. If T_1 contains at least one

[1] Characters as defined here are not elements of some alphabet but functions that assign an element of some alphabet to each taxon.

edge, subdivide an edge of T_1 with a new vertex u'; otherwise, set u' to be the single isolated vertex of T_1. Finally, obtain a binary X-tree T' by adding the new edge $\{u', v\}$. We say that T' has been obtained from T by a single *subtree prune and regraft (SPR)* operation. We next define a generalization of the SPR operation. Again, let e be an edge of T, and let T_1 and T_2 be the two trees obtained from T by deleting e and suppressing any resulting degree-2 vertices. For each $i \in \{1, 2\}$, if T_i has at least one edge, subdivide an edge in T_i with a new vertex v_i and, otherwise, set v_i to be the single vertex of T_i. Obtain a binary X-tree T' by adding the new edge $\{v_1, v_2\}$. We say that T' has been obtained from T by a single *tree bisection and reconnection (TBR)* operation.

NNI, *k*-ECR, and *k*-sECR. Let T be a binary X-tree. Let $e = \{u, v\}$ be an edge of T and let $e' = \{v, w\}$ be an internal edge of T that is adjacent to e. Let T' be a binary X-tree obtained from T by deleting e, suppressing v, subdividing an edge that is incident with w with a new vertex v', and joining u and v' via a new edge. We say that T' has been obtained from T by a single *nearest neighbor interchange (NNI)* operation. Equivalently, if T' is a binary refinement of the tree obtained from T by contracting e' and T' is non-isomorphic to T, then T' is obtained from T by a single NNI operation.

Now let T be a binary X-tree, and let k be a positive integer. Let T' be a binary refinement of a tree obtained from T by contracting k (distinct) internal edges E'. If T' and T are non-isomorphic, then we say that T' is a single *k-edge contract and refine (k-ECR)* operation [13] apart from T and that E' is a *contraction set* for T and T'. Note that an NNI operation is a 1-ECR operation and vice versa. We denote the restricted version of a k-ECR operation that requires the k contracted edges to form a subtree of T as k-sECR [29].

Distance measures. Let T and T' be binary X-trees. For each $\Theta \in \{\text{NNI}, \text{SPR}, \text{TBR}\}$, the distance $d_\Theta(T, T')$ is defined as the minimum number of Θ operations to transform T into T' [1]. The distance $d_{\text{ECR}}(T, T')$ is defined as the smallest number k such that T and T' are one k-ECR operation apart. Analogously, the distance $d_{\text{sECR}}(T, T')$ is defined as the smallest number k such that T and T' are one k-sECR operation apart.

Considered problems. In this work, we consider the parameterized complexity of the following problem for each distance measure $d \in \{d_{\text{NNI}}, d_{\text{SPR}}, d_{\text{TBR}}, d_{\text{ECR}}, d_{\text{sECR}}\}$.

d-LS Maximum Parsimony
Input: A set of taxa X, a binary X-tree T , a sequence of characters S, and an integer k.
Question: Is there a binary X-tree T' with $d(T, T') \leq k$ and $\text{score}_S(T') < \text{score}_S(T)$?

3 Properties of the Considered Distance Measures

In this section, we analyze the relation of the different distance measures.

▶ **Observation 3.1** ([1, 27]). *The distance measures d_{NNI}, d_{SPR}, and d_{TBR} are metrics.*

▶ **Lemma 3.2** (*). *The distance measure d_{ECR} is a metric.*

▶ **Observation 3.3.** *Let T and T' be distinct binary X-trees and let $k > 0$ be an integer. If $d_{\text{ECR}}(T, T') = k$, then there is a binary X-tree $\tilde{T}$ with $d_{\text{sECR}}(\tilde{T}, T') > 0$ such that $d_{\text{ECR}}(T, T') = d_{\text{ECR}}(T, \tilde{T}) + d_{\text{sECR}}(\tilde{T}, T')$.*

The idea behind Observation 3.3 is to consider the connected components of T induced by the contraction set S between T and T'. If S forms a subtree of T, then S is connected and $d_{\text{sECR}}(T,T') = d_{\text{ECR}}(T,T')$. Hence, the statement holds for $\tilde{T} = T'$. Otherwise, let $\tilde{S}$ be an inclusion-maximal subset of S, such that $\tilde{S}$ forms a subtree of T. Since $\tilde{S}$ is inclusion-maximal, we can obtain T' from T in two steps: First, we can obtain an intermediate X-tree $\tilde{T}$ from T by an sECR operation with contraction set $\tilde{S}$. Second, we can obtain T' from $\tilde{T}$ by an ECR operation with contraction set $S \setminus \tilde{S}$.

▶ **Lemma 3.4 (*).** *Let T and T' be binary X-trees. Then, $d_{\text{NNI}}(T,T') \geq d_{\text{ECR}}(T,T')$.*

▶ **Lemma 3.5.** *Let T and T' be binary X-trees. Then, $d_{\text{sECR}}(T,T') \geq d_{\text{SPR}}(T,T')$.*

Proof. Let $k = d_{\text{sECR}}(T,T')$. Hence, there is a set S of k internal edges in T such that T' can be obtained by an sECR operation with contraction set S. Let V' be the vertices of T incident with some edge of S and let V^* be the neighbors of V' in T that are not incident with any edge of S. Recall that by definition of sECR operations, the edges of S induce a subtree of T. Hence, $T(V^*)$ is a binary V^*-tree having the set S as internal edges. For each vertex v of V^*, let T_v denote the pendant subtree of T with pseudo-root v obtained by removing the edge between v and the unique neighbor of v in V'. Since T' can be obtained by an sECR operation with contraction set S, T' contains a subtree T'_v isomorphic to T_v for each vertex v of V^*. Hence, $d_{\text{SPR}}(T,T') = d_{\text{SPR}}(T_S,T'_S)$, where T_S is obtained from T by replacing T_v by the auxiliary taxa v for each vertex v of V^* and where T'_S is obtained from T' by replacing T'_v by the auxiliary taxa v for each vertex v of V^* [1]. Note that $T_S = T(V^*)$.

Hence, it remains to show that $d_{\text{SPR}}(T_S,T'_S) \leq k$. Since T is binary and the edges of S induce a subtree of T, $|V^*| = |S| + 3$. Moreover, since for each set of taxa X' and each two binary X'-trees $\tilde{T}$ and $\hat{T}$, $d_{\text{SPR}}(\tilde{T},\hat{T}) \leq |X'| - 3$ [1], we conclude $d_{\text{SPR}}(T_S,T'_S) \leq |V^*| - 3 = |S| = k$. Consequently, $d_{\text{SPR}}(T,T') \leq k = d_{\text{sECR}}(T,T')$. ◀

▶ **Lemma 3.6 (*).** *Let T and T' be binary X-trees. Then, $d_{\text{ECR}}(T,T') \geq d_{\text{SPR}}(T,T')$.*

4 Hardness of d-LS Maximum Parsimony

In this section, we establish our main theorem.

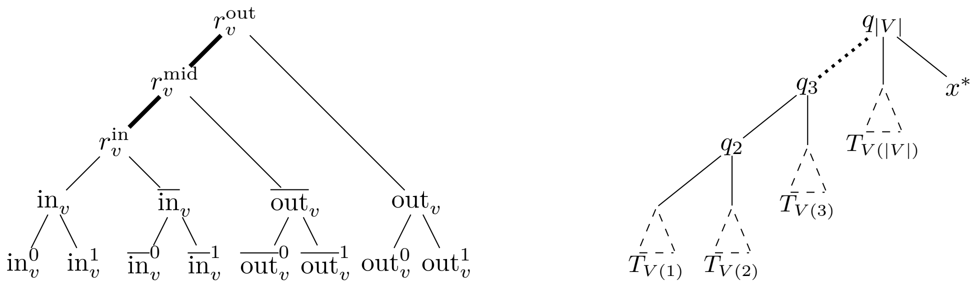
▶ **Theorem 4.1.** *For each distance measure $d \in \{d_{\text{NNI}}, d_{\text{ECR}}, d_{\text{SPR}}, d_{\text{TBR}}\}$ and even if each character is binary, d-LS MAXIMUM PARSIMONY*

- *is* NP*-complete,* W[1]*-hard when parameterized by k, and*
- *cannot be solved in $f(k) \cdot |I|^{o(k)}$ time for any computable function f, unless the ETH fails.*

We reduce from CLIQUE which is NP-hard [22], W[1]-hard when parameterized by k [7], and cannot be solved in $f(k) \cdot |I|^{o(k)}$ time for any computable function f, unless the ETH fails [5, 6].

CLIQUE
Input: An undirected graph $G = (V,E)$ and an integer k.
Question: Is there a *clique* of size k in G, that is, a set of vertices K of size k, such that $|E(K)| = \binom{k}{2}$?

Let $I = (G = (V,E),k)$ be an instance of CLIQUE and let $d \in \{d_{\text{NNI}}, d_{\text{ECR}}, d_{\text{SPR}}, d_{\text{TBR}}\}$ be a distance measure. We describe how to construct an equivalent instance $I' = (X, T = (V',E'), S, k')$ of d-LS MAXIMUM PARSIMONY in polynomial time where $k' := k$ if $d \in \{d_{\text{SPR}}, d_{\text{TBR}}\}$ and $k' := 2k$ if $d \in \{d_{\text{NNI}}, d_{\text{ECR}}\}$.



(a) For a vertex $v \in V$, the pendant X_v-tree T_v. The bold edges are the only edges of T_v that are not in R.

(b) The subtree of T connecting the pendant trees T_v for each vertex $v \in V$.

Figure 1 The construction of the X-tree T.

Definition of X and T. We start with an empty taxa set X and add for each vertex $v \in V$, a set X_v consisting of the eight taxa

$$\text{in}_v^0, \text{in}_v^1, \overline{\text{in}}_v^0, \overline{\text{in}}_v^1, \overline{\text{out}}_v^0, \overline{\text{out}}_v^1, \text{out}_v^0, \text{and out}_v^1$$

to X. Additionally, we add a taxon x^* to X. This completes the definition of X.

Next, we define the binary X-tree $T = (V', E')$. Since X contains $8 \cdot |V| + 1$ taxa and each internal vertex of T has three neighbors, T' has $16 \cdot |V|$ vertices and $2 \cdot |X| - 3 = 16 \cdot |V| - 1$ edges. By definition, V' is a superset of X. Additionally, for each vertex $v \in V$, the set V' contains the seven vertices

$$\text{in}_v, \overline{\text{in}}_v, \overline{\text{out}}_v, \text{out}_v, r_v^{\text{in}}, r_v^{\text{mid}}, \text{and } r_v^{\text{out}}.$$

The subtree $T_v := T(X_v)$ is depicted in Figure 1a.

Moreover, V' contains $|V| - 1$ additional vertices q_i with $i \in [2, |V|]$. Fix some arbitrary ordering of the vertices of V and let $V(i)$ denote the ith vertex of V according to that ordering. The vertex q_2 is adjacent to $r_{V(1)}^{\text{out}}, r_{V(2)}^{\text{out}}$, and q_3. For each $i \in [3, |V| - 1]$, the vertex q_i is adjacent to q_{i-1}, q_{i+1}, and $r_{V(i)}^{\text{out}}$. Finally, $q_{|V|}$ is adjacent to $q_{|V|-1}$, $r_{V(|V|)}^{\text{out}}$, and x^*. See Figure 1b for an illustration. This completes the definition of T.

Intuition. The idea of the reduction is as follows: Some of the characters that we define in the following will ensure that each binary X-tree T' that improves over T contains a pendant subtree $T'(X_v)$ for each vertex $v \in V$. Further characters will ensure that there are only two non-isomorphic trees for $T'(X_v)$ which are depicted in Figure 1a and Figure 2. Intuitively, these two choices then function as a selection gadget for selecting vertex v as a vertex of the sought clique K. The budget k' bounds how many such vertices can be selected. Finally, further characters will ensure that T' improves over T only if $E(K)$ contains at least $\binom{k}{2}$ edges.

Definition of the characters of S. Next, we define the characters of S which are all binary characters whose character states are 0 and 1. We obtain S by concatenating two sequences of characters, S_G and S_R, which we describe in the following.

First, we describe the characters of S_G. An overview of the characters is given in Table 1. We initialize S_G as the empty sequence.

For each edge $e \in E$, we add a character c_e to S_G. Let e be an edge of E. We set $c_e(x^*) := 1$. Let v be a vertex of V. If v is an endpoint of e, we set $c_e(x) := 1$ for each taxon $x \in \{\mathrm{in}_v^0, \mathrm{in}_v^1, \overline{\mathrm{in}}_v^0, \overline{\mathrm{in}}_v^1\}$ and we set $c_e(x) := 0$ for each taxon $x \in \{\overline{\mathrm{out}}_v^0, \overline{\mathrm{out}}_v^1, \mathrm{out}_v^0, \mathrm{out}_v^1\}$. Otherwise, if v is not an endpoint of e, we set $c_e(x) := 1$ for each taxon $x \in \{\mathrm{in}_v^1, \overline{\mathrm{in}}_v^1, \overline{\mathrm{out}}_v^1, \mathrm{out}_v^1\}$ and we set $c_e(x) := 0$ for each taxon $x \in \{\mathrm{in}_v^0, \overline{\mathrm{in}}_v^0, \overline{\mathrm{out}}_v^0, \mathrm{out}_v^0\}$. Let S_E denote the sequence of characters c_e for each edge $e \in E$.

Next, we define a character c_{mal}. We set $c_{\mathrm{mal}}(x^*) := 1$. For each vertex $v \in V$, we set $c_{\mathrm{mal}}(\mathrm{out}_v^0) = c_{\mathrm{mal}}(\mathrm{out}_v^1) := 1$ and we set $c_{\mathrm{mal}}(x) := 0$ for each taxon $x \in X_v \setminus \{\mathrm{out}_v^0, \mathrm{out}_v^1\}$. We add a sequence S_{mal} of $\binom{k}{2} - 1$ copies of c_{mal} to S_G. Intuitively, in a binary X-tree T', if both endpoints of an edge $e \in E$ are in the selected set K, then the parsimony score of c_e in T' is exactly the parsimony score of c_e in T minus one. Moreover, if T' is non-isomorphic to T, then the parsimony score of S_{mal} in T' is exactly the parsimony score of S_{mal} in T plus $|S_{\mathrm{mal}}|$. Hence, the characters of S_{mal} act as a hurdle to ensure that $E(K)$ contains at least $|S_{\mathrm{mal}}| + 1 = \binom{k}{2}$ edges.

Finally, for each vertex $v \in V$, we define four characters $c_{v,\mathrm{in}}$, $c_{v,\mathrm{out}}$, $c_{v,\mathrm{ri}}$, and $c_{v,\mathrm{ro}}$. For each taxon x of $X \setminus X_v$, we set $c_{v,\mathrm{in}}(x) := c_{v,\mathrm{out}}(x) := c_{v,\mathrm{ri}}(x) := c_{v,\mathrm{ro}}(x) := 1$. Now, let x be a taxon of X_v.

- If x is in $\{\mathrm{in}_v^0, \mathrm{in}_v^1\}$, we set $c_{v,\mathrm{in}}(x) := 1$, $c_{v,\mathrm{out}}(x) := 0$, $c_{v,\mathrm{ri}}(x) := 1$, and $c_{v,\mathrm{ro}}(x) := 0$.
- If x is in $\{\overline{\mathrm{in}}_v^0, \overline{\mathrm{in}}_v^1\}$, we set $c_{v,\mathrm{in}}(x) := 1$, $c_{v,\mathrm{out}}(x) := 0$, $c_{v,\mathrm{ri}}(x) := 0$, and $c_{v,\mathrm{ro}}(x) := 0$.
- If x is in $\{\overline{\mathrm{out}}_v^0, \overline{\mathrm{out}}_v^1\}$, we set $c_{v,\mathrm{in}}(x) := 0$, $c_{v,\mathrm{out}}(x) := 1$, $c_{v,\mathrm{ri}}(x) := 0$, and $c_{v,\mathrm{ro}}(x) := 0$.
- If x is in $\{\mathrm{out}_v^0, \mathrm{out}_v^1\}$, we set $c_{v,\mathrm{in}}(x) := 0$, $c_{v,\mathrm{out}}(x) := 1$, $c_{v,\mathrm{ri}}(x) := 0$, and $c_{v,\mathrm{ro}}(x) := 1$.

Let $\alpha := 2|X| \cdot (|E| + \binom{k}{2})$. Note that α is larger than $\mathrm{score}_{S_E}(T') + \mathrm{score}_{S_{\mathrm{mal}}}(T')$ of any binary X-tree T', since such a tree T' contains less than $2|X|$ edges and $|S_E| + |S_{\mathrm{mal}}| = |E| + \binom{k}{2} - 1$. For each vertex $v \in V$, we extend S_G by

- a sequence $S_{v,\mathrm{in}}$ of α copies of $c_{v,\mathrm{in}}$,
- a sequence $S_{v,\mathrm{out}}$ of α copies of $c_{v,\mathrm{out}}$,
- a sequence $S_{v,\mathrm{ri}}$ of 2α copies of $c_{v,\mathrm{ri}}$, and
- a sequence $S_{v,\mathrm{ro}}$ of 2α copies of $c_{v,\mathrm{ro}}$.

Let S_v denote the combined sequences of $S_{v,\mathrm{in}}$, $S_{v,\mathrm{out}}$, $S_{v,\mathrm{ri}}$, and $S_{v,\mathrm{ro}}$. Intuitively, for each binary X-tree T' that improves over T and contains $T'(X_v)$ as a pendant subtree, the characters of S_v ensure that $T'(X_v)$ is isomorphic to either the pendant tree depicted in Figure 1a or the pendant tree depicted in Figure 2. These two choices then function as a selection gadget for the vertices of the sought clique in G. This completes the construction of S_G. Note that $|S_G| = |E| + \binom{k}{2} - 1 + 6\alpha \cdot |V|$.

Next, we describe the sequence of characters S_R. Let $\beta := 2|X| \cdot |S_G|$. Note that β is larger than $\mathrm{score}_{S_G}(\tilde{T})$ of any binary X-tree $\tilde{T}$, since such a tree $\tilde{T}$ contains less than $2|X|$ edges. Let $R := E' \setminus \{\{r_v^{\mathrm{in}}, r_v^{\mathrm{mid}}\}, \{r_v^{\mathrm{mid}}, r_v^{\mathrm{out}}\} \mid v \in V\}$. For each edge e of R, we define a character c_R^e. Let $A|B$ be the split of T induced by e. For each taxon $x \in A$, we set $c_R^e(x) := 0$ and for each taxon $x \in B$, we set $c_R^e(x) := 1$. We add as sequence S_R^e of β copies of c_R^e to S_R. Intuitively, the characters of S_R ensure that each binary X-tree T' that improves over T, shares the split that is induced by e in T for each edge e of R. This implies that $T'(X_v)$ is a pendant subtree of T' for each vertex $v \in V$.

Properties of binary X-trees. Before we show the correctness of the reduction, we first make some observations about binary X-trees with the characters of the construction.

Note that for each binary X-tree T' and each edge e of R, $\mathrm{score}_{c_R^e}(T') \geq 1$.

Table 1 An overview of the characters of S_G.

	$c_e \in S_E$	$c_e \in S_E$	c_{mal}	$c_{v,\mathrm{in}}$	$c_{v,\mathrm{out}}$	$c_{v,\mathrm{ri}}$	$c_{v,\mathrm{ro}}$	$c \in S_w$
	$v \in e$	$v \notin e$						$w \neq v$
x^*	1	1	1	1	1	1	1	1
in_v^0	1	0	0	1	0	1	0	1
in_v^1	1	1	0	1	0	1	0	1
$\overline{\mathrm{in}}_v^0$	1	0	0	1	0	0	0	1
$\overline{\mathrm{in}}_v^1$	1	1	0	1	0	0	0	1
$\overline{\mathrm{out}}_v^0$	0	0	0	0	1	0	0	1
$\overline{\mathrm{out}}_v^1$	0	1	0	0	1	0	0	1
out_v^0	0	0	1	0	1	0	1	1
out_v^1	0	1	1	0	1	0	1	1

▶ **Definition 4.2.** *Let T' be a binary X-tree. We say that T' is* split-consistent for T and R *if for each edge e of R, the split of T induced by e is also a split of T'.*

In preparation for the next observation, note that if a binary X-tree T' is not split-consistent for T and R, then there is some edge e of R such that $\mathrm{score}_{c^e_R}(T') \geq 2$ and thus $\mathrm{score}_{S^e_R}(T') \geq 2 \cdot \beta$. Hence, $\mathrm{score}_S(T') \geq \mathrm{score}_{S_R}(T') \geq \beta \cdot (|R| + 1)$. Since $\beta > \mathrm{score}_{S_G}(T)$, this implies $\mathrm{score}_S(T') > \mathrm{score}_S(T)$. Hence, we conclude the following.

▶ **Observation 4.3.** *Let T' be a binary X-tree. a) If $\mathrm{score}_S(T') \leq \mathrm{score}_S(T)$, then T' is split-consistent for T and R. b) If T' is split-consistent for T and R, then $\mathrm{score}_{S_R}(T') = \beta \cdot |R|$.*

To determine whether I' is a yes-instance of d-LS Maximum Parsimony, we analyze the structure of binary X-trees T' with $\mathrm{score}_S(T') \leq \mathrm{score}_S(T)$. Due to Observation 4.3, we only need to consider binary X-trees that are split-consistent for T and R in the following.

Let v be a vertex of V and let T' be a binary X-tree which is split-consistent for T and R. Since there is an edge e_v in T such that e_v induces the split $X_v|(X \setminus X_v)$ in T and e_v is contained in R, $X_v|(X \setminus X_v)$ is a split in T'. Hence, $T'(X_v)$ is a pendant tree. Moreover, since all edges incident with in_v are in R, we can assume that in_v is the common neighbor of in_v^0 and in_v^1 in T'. Similarly, we may assume that $\overline{\mathrm{in}}_v$ is the common neighbor of $\overline{\mathrm{in}}_v^0$ and $\overline{\mathrm{in}}_v^1$ in T', $\overline{\mathrm{out}}_v$ is the common neighbor of $\overline{\mathrm{out}}_v^0$ and $\overline{\mathrm{out}}_v^1$ in T', and out_v is the common neighbor of out_v^0 and out_v^1 in T'.

▶ **Definition 4.4.** *Let T' be a binary X-tree which is split-consistent for T and R, let v be a vertex of V, and let r be the pseudo-root of the pendant tree $T'(X_v)$. We say that $T'(X_v)$ is an* in-rooting *of T_v if in_v is adjacent to r, $\overline{\mathrm{in}}_v$ has distance 2 to r, and both $\overline{\mathrm{out}}_v$ and out_v have distance 3 to r. Similarly, we say that $T'(X_v)$ is an* out-rooting *of T_v if out_v is adjacent to r, $\overline{\mathrm{out}}_v$ has distance 2 to r, and both $\overline{\mathrm{in}}_v$ and in_v have distance 3 to r.*

Figure 1a shows an out-rooting of T_v and Figure 2 shows an in-rooting of T_v.

Note that for each vertex v of V, there is a unique in-rooting of T_v with respect to isomorphism. Similarly, there is a unique out-rooting of T_v with respect to isomorphism. Note that for each vertex $v \in V$, T_v is an out-rooting of T_v. We call a binary X-tree T' *well-rooted* if T' is split-consistent for T and R and if for each vertex $v \in V$, $T'(X_v)$ is either an in-rooting or an out-rooting of T_v. Note that T is well-rooted.

▶ **Lemma 4.5 (*).** *Let T' be a binary X-tree which is split-consistent for T and R and let v be a vertex of V. If $T'(X_v)$ is an in-rooting of T_v or an out-rooting of T_v, then $\mathrm{score}_{S_v}(T') = 9\alpha$. Otherwise, $\mathrm{score}_{S_v}(T') \geq 10\alpha$.*

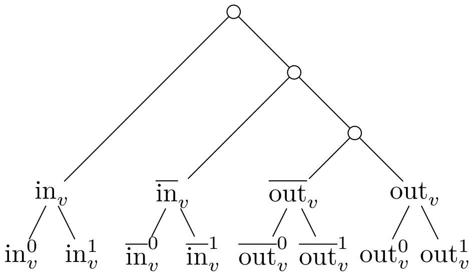

in_v
$\overline{\mathrm{in}}_v$
$\overline{\mathrm{out}}_v$
out_v
in_v^0
in_v^1
$\overline{\mathrm{in}}_v^0$
$\overline{\mathrm{in}}_v^1$
$\overline{\mathrm{out}}_v^0$
$\overline{\mathrm{out}}_v^1$
out_v^0
out_v^1

Figure 2 An in-rooting of T_v.

Next, we describe for a given well-rooted binary X-tree T' the maximum parsimony scores of T' with respect to the characters of S_E and S_{mal}. The idea is that in a well-rooted binary X-tree T', for each edge $e = \{u, v\} \in E$ where $T'(X_u)$ is an in-rooting of T_u and where $T'(X_v)$ is an in-rooting of T_v, the parsimony score of the character c_e in T' is exactly the parsimony score of the character c_e in T minus one. Moreover, if $T'(X_v)$ is an in-rooting of T_v for at least one vertex $v \in V$, then the parsimony score of the characters of S_{mal} in T' is exactly the parsimony score of the characters of S_{mal} in T plus $\binom{k}{2} - 1$.

▶ **Lemma 4.6.** *Let T' be a well-rooted binary X-tree. Let $e = \{u, v\}$ be an edge of E.*

a) *If $T'(X_u)$ is an in-rooting of T_u and $T'(X_v)$ is an in-rooting of T_v, then* $\mathrm{score}_{c_e}(T') = 4(|V| - 2) + 2$. *Otherwise,* $\mathrm{score}_{c_e}(T') = 4(|V| - 2) + 3$.

b) *If there is a vertex $w \in V$ such that $T'(X_w)$ is an in-rooting of T_w, then* $\mathrm{score}_{c_{\mathrm{mal}}}(T') = |V| + 1$. *Otherwise, that is, if T' is isomorphic to T,* $\mathrm{score}_{c_{\mathrm{mal}}}(T') = |V|$.

Proof. For each vertex w of V, let $T'_w := T'(X_w)$. Let V_{in} be those vertices w of V, where T'_w is an in-rooting of T_w and let $V_{\mathrm{out}} = V \setminus V_{\mathrm{in}}$ be those vertices w of V, where T'_w is an out-rooting of T_w. For each vertex $w \in V_{\mathrm{in}}$, let r_w^{in} be the name of the pseudo-root of T'_w, let r_w^{mid} and in_w be the neighbors of r_w^{in}, and let r_w^{out} and $\overline{\mathrm{in}}_w$ be the neighbors of r_w^{mid}. Analogously, for each vertex $w \in V_{\mathrm{out}}$, let r_w^{out} be the name of the pseudo-root of T'_w, let r_w^{mid} and out_w be the neighbors of r_w^{out}, and let r_w^{in} and $\overline{\mathrm{out}}_w$ be the neighbors of r_w^{mid}. Recall that since T' is well-rooted, for each vertex $w \in V$, in_w is adjacent to both in_w^0 and in_w^1, $\overline{\mathrm{in}}_w$ is adjacent to both $\overline{\mathrm{in}}_w^0$ and $\overline{\mathrm{in}}_w^1$, $\overline{\mathrm{out}}_w$ is adjacent to both $\overline{\mathrm{out}}_w^0$ and $\overline{\mathrm{out}}_w^1$, and out_w is adjacent to both out_w^0 and out_w^1.

First, we show statement a). Let c_e be a character for some edge $e = \{u, v\}$ of E. For each vertex $w \in V \setminus \{u, v\}$,

- let P_{in_w} be the unique path between in_w^0 and in_w^1 in T',
- let $P_{\overline{\mathrm{in}}_w}$ be the unique path between $\overline{\mathrm{in}}_w^0$ and $\overline{\mathrm{in}}_w^1$ in T',
- let $P_{\overline{\mathrm{out}}_w}$ be the unique path between $\overline{\mathrm{out}}_w^0$ and $\overline{\mathrm{out}}_w^1$ in T', and
- let P_{out_w} be the unique path between out_w^0 and out_w^1 in T'.

Note that each of these four paths only contains two edges and that these four paths are pairwise edge-disjoint. Let $\mathcal{P}_w := \{P_{\mathrm{in}_w}, P_{\overline{\mathrm{in}}_w}, P_{\overline{\mathrm{out}}_w}, P_{\mathrm{out}_w}\}$. Let P' be a path in $\mathcal{P}_w$ and let w^0 and w^1 be the terminals of P'. Since by definition $c_e(w^0) \neq c_e(w^1)$, for each extension c_e^* of c_e in T' at least one edge of P' is a mutation edge of c_e^*. Note that each path in $\mathcal{P}_w$ is edge-disjoint with each path in $\mathcal{P}_{w'}$ for distinct vertices w and w' of $V \setminus \{u, v\}$. Moreover, let P_u be the path between $\overline{\mathrm{in}}_u^0$ and $\overline{\mathrm{out}}_u^0$ in T' and let P_v be the path between $\overline{\mathrm{in}}_v^0$ and $\overline{\mathrm{out}}_v^0$ in T'. Note that P_u and P_v are edge-disjoint and that both are edge-disjoint with each path $P_w \in \mathcal{P}_w$ for each vertex $w \in V \setminus \{u, v\}$. Since $c_e(\overline{\mathrm{in}}_u^0) = 0$ and $c_e(\overline{\mathrm{out}}_u^0) = 1$, for each extension c_e^* of c_e in T', at least one edge of P_u is a mutation edge of c_e^*. Similarly, since $c_e(\overline{\text{in}}_v^0) = 0$ and $c_e(\overline{\text{out}}_v^0) = 1$, for each extension c_e^* of c_e in T', at least one edge of P_v is a mutation edge of c_e^*. Hence, $\text{score}_{c_e}(T') \geq 4(|V| - 2) + 2$.

Case 1: T_u' is an in-rooting of T_u and T_v' is an in-rooting of T_v. We define an extension c_e^* of c_e in T', such that $\text{score}_{c_e^*}(T') = 4(|V|-2)+2$. We set $c_e^*(\text{out}_u) := c_e^*(\overline{\text{out}}_u) := c_e^*(r_u^{\text{out}}) := 0$ and $c_e^*(\text{out}_v) := c_e^*(\overline{\text{out}}_v) := c_e^*(r_v^{\text{out}}) := 0$. For each remaining internal vertex v' of T', we set $c_e^*(v') := 1$. Hence, the edge set

$$\{\{r_u^{\text{out}}, r_u^{\text{mid}}\}, \{r_v^{\text{out}}, r_v^{\text{mid}}\}\} \\ \cup \{\{\text{in}_w^0, \text{in}_w\}, \{\overline{\text{in}}_w^0, \overline{\text{in}}_w\}, \{\overline{\text{out}}_w^0, \overline{\text{out}}_w\}, \{\text{out}_w^0, \text{out}_w\} \mid w \in V \setminus \{u, v\}\}$$

contains the mutation edges of c_e^* in T'. Consequently, $\text{score}_{c_e^*}(T') = 4(|V| - 2) + 2$ which implies $\text{score}_{c_e}(T') = 4(|V| - 2) + 2$.

Case 2: T_u' is an out-rooting of T_u or T_v' is an out-rooting of T_v. Assume without loss of generality that T_v' is an out-rooting of T_v. Let P_x^* be the unique path between out_v^0 and x^* in T'. Since $c_e(\text{out}_v^0) = 0$ and $c_e(x^*) = 1$, for each extension c_e^* of c_e in T', at least one edge of P_x^* is a mutation edge of c_e^*. Note that P_x^* is edge-disjoint with P_u and edge-disjoint with each path $P_w \in \mathcal{P}_w$ for each vertex $w \in V \setminus \{u, v\}$. Moreover, since T_v' is an out-rooting of T_v, P_x^* is also edge-disjoint with P_v. Hence, $\text{score}_{c_e}(T') \geq 4(|V| - 2) + 3$. We define an extension c_e^* of c_e in T', such that $\text{score}_{c_e^*}(T') = 4(|V| - 2) + 3$. To this end, we distinguish whether T_u' is an in-rooting of T_u or an out-rooting of T_u.

Case 2.1: T_u' is an in-rooting of T_u. We set $c_e^*(\text{out}_u) := c_e^*(\overline{\text{out}}_u) := c_e^*(r_u^{\text{out}}) := 0$ and $c_e^*(\text{out}_v) := c_e^*(\overline{\text{out}}_v) := 0$. For each remaining internal vertex v' of T', we set $c_e^*(v') := 1$. Hence, the edge set

$$\{\{r_u^{\text{out}}, r_u^{\text{mid}}\}, \{r_v^{\text{mid}}, \overline{\text{out}}_v\}, \{r_v^{\text{out}}, \text{out}_v\}\} \\ \cup \{\{\text{in}_w^0, \text{in}_w\}, \{\overline{\text{in}}_w^0, \overline{\text{in}}_w\}, \{\overline{\text{out}}_w^0, \overline{\text{out}}_w\}, \{\text{out}_w^0, \text{out}_w\} \mid w \in V \setminus \{u, v\}\}$$

contains the mutation edges of c_e^* in T'.

Case 2.2: T_u' is an out-rooting of T_u. We set $c_e^*(\text{in}_u) := c_e^*(\overline{\text{in}}_u) := c_e^*(r_u^{\text{in}}) := 1$ and $c_e^*(\text{in}_v) := c_e^*(\overline{\text{in}}_v) := c_e^*(r_v^{\text{in}}) := 1$. For each remaining internal vertex v' of T', we set $c_e^*(v') := 0$. Hence, the edge set

$$\{\{r_u^{\text{in}}, r_u^{\text{mid}}\}, \{r_v^{\text{in}}, r_v^{\text{mid}}\}, \{x^*, q_n\}\} \\ \cup \{\{\text{in}_w^1, \text{in}_w\}, \{\overline{\text{in}}_w^1, \overline{\text{in}}_w\}, \{\overline{\text{out}}_w^1, \overline{\text{out}}_w\}, \{\text{out}_w^1, \text{out}_w\} \mid w \in V \setminus \{u, v\}\}$$

contains the mutation edges of c_e^* in T'.

Consequently, in both cases $\text{score}_{c_e^*}(T') = 4(|V| - 2) + 3$ which implies $\text{score}_{c_e}(T') = 4(|V| - 2) + 3$.

Next, we show statement b). Consider the character c_{mal}. For each vertex $v \in V$, let P_v be the unique path between $\overline{\text{out}}_v^0$ and out_v^0 in T'. Since $c_{\text{mal}}(\overline{\text{out}}_v^0) = 0$ and $c_{\text{mal}}(\text{out}_v^0) = 1$, for each extension c_{mal}^* of c_{mal} in T' at least one edge of P_v is a mutation edge of c_{mal}^*. Note that the paths P_v and P_w are edge-disjoint for distinct vertices v and w of V. Hence, $\text{score}_{c_{\text{mal}}}(T') \geq |V|$.

Case 1: There is some vertex $v \in V$ such that T'_v is an in-rooting of T_v. Let P^*_x be the unique path between in^0_v and x^* in T'. Since $c_{\mathrm{mal}}(\mathrm{in}^0_v) = 0$ and $c_{\mathrm{mal}}(x^*) = 1$, for each extension c^*_{mal} of c_{mal} in T', at least one edge of P^*_x is a mutation edge of c^*_{mal}. Note that P^*_x is edge-disjoint with P_w for each vertex $w \in V$ distinct from v. Moreover, since T'_v is an in-rooting of T_v, P^*_x is also edge-disjoint with P_v. Hence, $\mathrm{score}_{c_{\mathrm{mal}}}(T') \geq |V| + 1$. We define an extension c^*_{mal} of c_{mal} in T', such that $\mathrm{score}_{c^*_{\mathrm{mal}}}(T') = |V| + 1$. We set $c^*_{\mathrm{mal}}(\mathrm{out}_w) := 1$, for each vertex $w \in V$. For each remaining internal vertex v' of T', we set $c^*_{\mathrm{mal}}(v') := 0$. Hence, the edge set $\{\{q_n, x^*\}\} \cup \{\{\mathrm{out}_v, r^{\mathrm{out}}_v\} \mid v \in V\}$ contains the mutation edges of c^*_{mal} in T'. Consequently, $\mathrm{score}_{c^*_{\mathrm{mal}}}(T') = |V| + 1$ which implies $\mathrm{score}_{c_{\mathrm{mal}}}(T') = |V| + 1$.

Case 2: For each vertex $v \in V$, T'_v is an out-rooting of T_v. Hence, T' is isomorphic to T. We define an extension c^*_{mal} of c_{mal} in T', such that $\mathrm{score}_{c^*_{\mathrm{mal}}}(T') = |V|$. We set $c^*_{\mathrm{mal}}(\mathrm{in}_v) := c^*_{\mathrm{mal}}(\overline{\mathrm{in}}_v) := c^*_{\mathrm{mal}}(\mathrm{out}_v) := c^*_{\mathrm{mal}}(r^{\mathrm{in}}_v) := c^*_{\mathrm{mal}}(r^{\mathrm{mid}}_v) := 0$, for each vertex $v \in V$. For each remaining internal vertex v' of T', we set $c^*_{\mathrm{mal}}(v') = 1$. Hence, the edge set $\{\{r^{\mathrm{mid}}_v, r^{\mathrm{out}}_v\} \mid v \in V\}$ contains the mutation edges of c^*_{mal} in T'. Consequently, $\mathrm{score}_{c^*_{\mathrm{mal}}}(T') = |V|$ which implies that $\mathrm{score}_{c_{\mathrm{mal}}}(T') = |V|$. ◀

The score of improving X-trees with respect to S. Since T is well-rooted, and for each vertex $v \in V$, T_v is an out-rooting of T_v, Observation 4.3, Lemma 4.5, and Lemma 4.6 imply the following.

▶ **Corollary 4.7.** $\mathrm{score}_S(T) = |E| \cdot (4(|V| - 2) + 3) + (\binom{k}{2} - 1) \cdot |V| + |V| \cdot 9\alpha + |R| \cdot \beta$.

Note that by definition, $\alpha = 2(8|V| + 1) \cdot (|E| + \binom{k}{2}) > |E| \cdot (4(|V| - 2) + 3) + (\binom{k}{2} - 1) \cdot |V|$. Hence, $\mathrm{score}_S(T) < \alpha \cdot (9|V| + 1) + |R| \cdot \beta$.

▶ **Corollary 4.8.** *Let T' be a binary X-tree with $\mathrm{score}_S(T') < \mathrm{score}_S(T)$. Then, T' is well-rooted.*

Proof. Due to Observation 4.3, T' is split-consistent for T and R and $\mathrm{score}_{S_R}(T') = |R| \cdot \beta$. Assume towards a contradiction that there is a vertex $v \in V$ such that $T'(X_v)$ is neither an in-rooting of T_v nor an out-rooting of T_v. Hence, Lemma 4.5 implies $\mathrm{score}_{S_v}(T') \geq 10\alpha$ and $\mathrm{score}_{S_w}(T') \geq 9\alpha$ for each vertex $w \in V \setminus \{v\}$. Consequently, $\mathrm{score}_S(T') \geq 10\alpha + (|V| - 1) \cdot 9\alpha + |R| \cdot \beta = \alpha \cdot (9|V| + 1) + |R| \cdot \beta > \mathrm{score}_S(T)$, a contradiction. ◀

Distances between well-rooted binary X-trees. Next, we describe for each distance measure $d \in \{d_{\mathrm{NNI}}, d_{\mathrm{ECR}}, d_{\mathrm{SPR}}, d_{\mathrm{TBR}}\}$ the distance between T and any other well-rooted binary X-tree T'.

▶ **Lemma 4.9.** *Let T' be a binary and well-rooted X-tree. Moreover, let K be the set of vertices of V such that $T'(X_v)$ is an in-rooting of T_v for each vertex $v \in K$ and $T'(X_w)$ is an out-rooting of T_w for each vertex $w \in V \setminus K$. Then, $d_{\mathrm{NNI}}(T, T') = d_{\mathrm{ECR}}(T, T') = 2 \cdot |K|$ and $d_{\mathrm{SPR}}(T, T') = d_{\mathrm{TBR}}(T, T') = |K|$.*

Proof. First, we show that $d_{\mathrm{NNI}}(T, T') = d_{\mathrm{ECR}}(T, T') = 2 \cdot |K|$. To this end, we show that $d_{\mathrm{NNI}}(T, T') \leq 2 \cdot |K|$ and that $d_{\mathrm{ECR}}(T, T') \geq 2 \cdot |K|$. Since $d_{\mathrm{NNI}}(T, T') \geq d_{\mathrm{ECR}}(T, T')$ due to Lemma 3.4, this then implies $d_{\mathrm{NNI}}(T, T') = d_{\mathrm{ECR}}(T, T') = 2 \cdot |K|$.

To show that $d_{\mathrm{NNI}}(T, T') \leq 2 \cdot |K|$, we prove the following: Let $\tilde{T}$ be a well-rooted binary X-tree and let v be a vertex such that $\tilde{T}(X_v)$ is an out-rooting of T_v. Then, $d_{\mathrm{NNI}}(\tilde{T}, \hat{T}) \leq 2$, where $\hat{T}$ is a well-rooted binary X-tree with $\tilde{T}(X \setminus X_v) = \hat{T}(X \setminus X_v)$ and where $\hat{T}(X_v)$

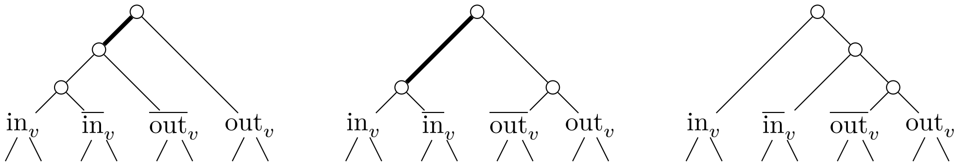


Figure 3 The two consecutive NNI operation transforming an out-rooting into an in-rooting.

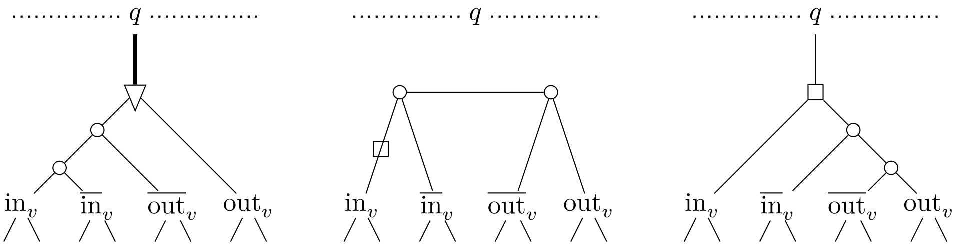


Figure 4 Transforming an out-rooting into an in-rooting by an SPR operation. First, the bold edge is removed and the triangular vertex is suppressed. Second, the unique internal edge incident with in_v is subdivided by the rectangular vertex. Finally, the rectangular vertex is joined with q by a new edge.

is an in-rooting of T_v. To show the claim, we describe two consecutive NNI operations transforming $\tilde{T}$ into $\hat{T}$. See Figure 3 for an illustration of these NNI operations. Let r_v^{out} be name of the pseudo-root of the pendant tree $\tilde{T}(X_v)$, let r_v^{out} be the name of the common neighbor of r_v^{mid} and out_v in $\tilde{T}$, and let r_v^{mid} be the name of the common neighbor of r_v^{in} and $\overline{\text{out}}_v$ in $\tilde{T}$. Moreover, let q be the unique neighbor of r_v^{out} outside of $\tilde{T}(X_v)$ in $\tilde{T}$. We obtain the well-rooted binary X-tree $\hat{T}$ from $\tilde{T}$ by

- firstly removing the edges $\{q, r_v^{\text{out}}\}$ and $\{\overline{\text{out}}_v, r_v^{\text{mid}}\}$ and adding the edges $\{\overline{\text{out}}_v, r_v^{\text{out}}\}$ and $\{q, r_v^{\text{mid}}\}$, and
- secondly removing the edges $\{q, r_v^{\text{mid}}\}$ and $\{\overline{\text{in}}_v, r_v^{\text{in}}\}$ and adding the edges $\{\overline{\text{in}}_v, r_v^{\text{mid}}\}$ and $\{q, r_v^{\text{in}}\}$.

Since this can be done by two consecutive NNI operations and $\tilde{T}(X \setminus X_v) = \hat{T}(X \setminus X_v)$, we conclude $d_{\text{NNI}}(\tilde{T}, \hat{T}) \leq 2$. Since d_{NNI} is a metric one can then show via induction over any arbitrary ordering of the vertices of K, that $d_{\text{NNI}}(T, T') \leq 2 \cdot |K|$.

It remains to show that $d_{\text{ECR}}(T, T') \geq 2 \cdot |K|$. Let $\tilde{E}$ be a subset of the internal edges of T, such that T' can be obtained from T by an ECR operation with contraction set $\tilde{E}$. We show that $|\tilde{E}| \geq 2 \cdot |K|$. Let v be a vertex of K. Recall that T_v is an out-rooting of T_v and that T'_v is an in-rooting of T_v. Hence, the edge $\{r_v^{\text{out}}, r_v^{\text{mid}}\}$ induces the split $A|B$ in T with $A := \{\text{in}_v^0, \text{in}_v^1, \overline{\text{in}}_v^0, \overline{\text{in}}_v^1, \overline{\text{out}}_v^0, \overline{\text{out}}_v^1\}$ and $B := X \setminus A$. Since $A|B$ is not a split of T', the edge $\{r_v^{\text{out}}, r_v^{\text{mid}}\}$ is contained in $\tilde{E}$. Similar, since the edge $\{r_v^{\text{mid}}, r_v^{\text{in}}\}$ induces the split $A|B$ in T with $A := \{\text{in}_v^0, \text{in}_v^1, \overline{\text{in}}_v^0, \overline{\text{in}}_v^1\}$ and $B := X \setminus A$. Since $A|B$ is not a split of T', the edge $\{r_v^{\text{mid}}, r_v^{\text{in}}\}$ is contained in $\tilde{E}$. Hence, for each vertex v of V, $\tilde{E}$ contains at least two edges of $T(X_v)$. Consequently, $|\tilde{E}| \geq 2 \cdot |K|$ which implies $d_{\text{ECR}}(T, T') \geq 2 \cdot |K|$.

Second, we show that $d_{\text{SPR}}(T, T') = d_{\text{TBR}}(T, T') = |K|$. Similar to the first part of the proof, we show that $d_{\text{SPR}}(T, T') \leq |K|$ and that $d_{\text{TBR}}(T, T') \geq |K|$. Since $d_{\text{SPR}}(T, T') \geq d_{\text{TBR}}(T, T')$ this then implies $d_{\text{SPR}}(T, T') = d_{\text{TBR}}(T, T') = |K|$.

To show that $d_{\text{SPR}}(T, T') \leq |K|$, we prove the following: Let $\tilde{T}$ be a well-rooted binary X-tree and let v be a vertex such that $\tilde{T}(X_v)$ is an out-rooting of T_v. Then, $d_{\text{SPR}}(\tilde{T}, \hat{T}) \leq 1$, where $\hat{T}$ is a well-rooted binary X-tree with $\tilde{T}(X \setminus X_v) = \hat{T}(X \setminus X_v)$ and where $\hat{T}(X_v)$ is an in-rooting of T_v.

To show this claim, we describe an SPR operation transforming $\tilde{T}$ into $\hat{T}$. See Figure 4 for an illustration of this SPR operation. Let r_v^{out} be the name of the pseudo-root of the pendant tree $\tilde{T}(X_v)$ and let q be the name of the unique neighbor of r_v^{out} outside of $\tilde{T}(X_v)$ in $\tilde{T}$. Moreover, let r_v^{in} be the name of the common neighbor of in_v and $\overline{\text{in}}_v$ in $\tilde{T}$. We obtain the well-rooted binary X-tree $\hat{T}$ from $\tilde{T}$ by: removing the edge $\{r_v^{\text{out}}, q\}$, suppressing the vertex r_v^{out}, subdividing the edge $\{\text{in}_v, r_v^{\text{in}}\}$ by a vertex q', and adding the edge $\{q, q'\}$. Since this can be done by a single SPR operation and $\tilde{T}(X \setminus X_v) = \hat{T}(X \setminus X_v)$, we conclude $d_{\text{SPR}}(\tilde{T}, \hat{T}) \leq 1$. Since d_{SPR} is a metric, one can then show via induction over any arbitrary ordering of the vertices of K, that $d_{\text{SPR}}(T, T') \leq |K|$.

It remains to show that $d_{\text{TBR}}(T, T') \geq |K|$. This proof is deferred to a full version of the article. ◀

Correctness. Finally, we are able to show that I is a yes-instance of Clique if and only if I' is a yes-instance of d-LS Maximum Parsimony with appropriate distance bounds.

▶ **Lemma 4.10.** *The following statements are equivalent:*

1. *There is a clique of size k in G.*
2. *There is a binary X-tree T' with $\text{score}_S(T') < \text{score}_S(T)$ and $d_{\text{SPR}}(T, T') \leq k$.*
3. *There is a binary X-tree T' with $\text{score}_S(T') < \text{score}_S(T)$ and $d_{\text{TBR}}(T, T') \leq k$.*
4. *There is a binary X-tree T' with $\text{score}_S(T') < \text{score}_S(T)$ and $d_{\text{NNI}}(T, T') \leq 2k$.*
5. *There is a binary X-tree T' with $\text{score}_S(T') < \text{score}_S(T)$ and $d_{\text{ECR}}(T, T') \leq 2k$.*

Proof. First, we show that Item 1 implies each of Item 2–5. Let $K \subseteq V$ be a clique of size k in G. Further, let T' be a well-rooted binary X-tree such that for each vertex $v \in K$, $T'(X_v)$ is an in-rooting of T_v, and for each vertex $v \in V \setminus K$, $T'(X_v)$ is an out-rooting of T_v. Due to Lemma 4.9, $d_{\text{SPR}}(T, T') = d_{\text{TBR}}(T, T') = k$ and $d_{\text{NNI}}(T, T') = d_{\text{ECR}}(T, T') = 2k$. It remains to show that $\text{score}_S(T') < \text{score}_S(T)$. Since T' is well-rooted, due to Observation 4.3, $\text{score}_{S_R}(T') = |R| \cdot \beta$ and due to Lemma 4.5, for each vertex $v \in V$, $\text{score}_{S_v}(T') = 9\alpha$. Moreover, since K is non-empty, we obtain by Lemma 4.6, that $\text{score}_{S_{\text{mal}}}(T') = (\binom{k}{2} - 1) \cdot (|V| + 1)$. Since K is a clique in G, $|E(K)| = \binom{k}{2}$. Finally, by Lemma 4.6, for each edge e of $E(K)$, $\text{score}_{c_e}(T') = 4(|V| - 2) + 2$, and for each edge e of $E \setminus E(K)$, $\text{score}_{c_e}(T') = 4(|V| - 2) + 3$. We conclude

$$\begin{aligned}\text{score}_S(T') &= |E| \cdot (4(|V| - 2) + 3) - \binom{k}{2} + \left(\binom{k}{2} - 1\right) \cdot (|V| + 1) + |V| \cdot 9\alpha + |R| \cdot \beta \\ &= |E| \cdot (4(|V| - 2) + 3) + \left(\binom{k}{2} - 1\right) \cdot |V| + |V| \cdot 9\alpha + |R| \cdot \beta - 1 = \text{score}_S(T) - 1,\end{aligned}$$

due to Corollary 4.7. Hence, T' is a binary X-tree with $\text{score}_S(T') < \text{score}_S(T)$, $d_{\text{SPR}}(T, T') = d_{\text{TBR}}(T, T') = k$, and $d_{\text{NNI}}(T, T') = d_{\text{ECR}}(T, T') = 2k$.

Second, we show that each of Item 2–5 implies Item 1. Let T' be a binary X-tree with a) $\text{score}_S(T') < \text{score}_S(T)$ and b) $d_{\text{SPR}}(T, T') \leq k$, $d_{\text{TBR}}(T, T') \leq k$, $d_{\text{NNI}}(T, T') \leq 2k$, or $d_{\text{ECR}}(T, T') \leq 2k$. Since $\text{score}_S(T') < \text{score}_S(T)$, due to Corollary 4.8, T' is well-rooted, that is, for each vertex $v \in V$, $T'_v := T'(X_v)$ is either an in-rooting of T_v or an out-rooting of T_v. Let $K \subseteq V$ be the set of all vertices v of V where T'_v is an in-rooting of T_v. We show that K is a clique of size k in G. Since $d_{\text{SPR}}(T, T') \leq k$, $d_{\text{TBR}}(T, T') \leq k$, $d_{\text{NNI}}(T, T') \leq 2k$, or $d_{\text{ECR}}(T, T') \leq 2k$, Lemma 4.9 implies that K has size at most k. Moreover, since $\text{score}_S(T') < \text{score}_S(T)$, T' is not isomorphic to T, which implies that K is nonempty. Hence due to Lemma 4.6, $\text{score}_{S_{\text{mal}}}(T') = (\binom{k}{2} - 1) \cdot (|V| + 1)$. Moreover, since T' is well-rooted, due to Observation 4.3, $\text{score}_{S_R}(T') = |R| \cdot \beta$ and due to Lemma 4.5, for each vertex $v \in V$, $\text{score}_{S_v}(T') = 9\alpha$. Finally, by Lemma 4.6, for each edge $e \in E \setminus E(K)$, $\text{score}_{c_e}(T') = 4(|V|-2)+3$, and for each edge $e \in E(K)$, $\text{score}_{c_e}(T') = 4(|V|-2)+2$. Consequently, $\text{score}_S(T) - \text{score}_S(T') = |E(K)| - (\binom{k}{2} - 1)$.

Since $\text{score}_S(T') < \text{score}_S(T)$, we have $|E(K)| \geq \binom{k}{2}$. Hence, K is a size-k clique in G. ◀

Since $k' = k$ if $d \in \{d_{\text{SPR}}, d_{\text{TBR}}\}$ and $k' = 2k$ if $d \in \{d_{\text{NNI}}, d_{\text{ECR}}\}$, Lemma 4.10 implies that I is a yes-instance of Clique if and only if I' is a yes-instance of d-LS Maximum Parsimony. This completes the proof of Theorem 4.1.

5 Essentially Tight Brute-Force Algorithms

We now show that simple brute-force algorithms for d-LS Maximum Parsimony for each distance measure $d \in \{d_{\text{NNI}}, d_{\text{ECR}}, d_{\text{SPR}}, d_{\text{TBR}}\}$ essentially match the lower bounds shown in Theorem 4.1. First, consider a distance measure $d \in \{d_{\text{NNI}}, d_{\text{SPR}}, d_{\text{TBR}}\}$.

▶ **Observation 5.1.** *Let T be a binary X-tree, let $d \in \{d_{\text{NNI}}, d_{\text{SPR}}, d_{\text{TBR}}\}$ be a distance measure, and let k be an integer. One can enumerate all binary X-trees T' with $d(T, T') \leq k$ in $|X|^{\mathcal{O}(k)}$ time.*

Observation 5.1 can be seen as follows: there are $|X|^{\mathcal{O}(1)}$ many binary X-trees T' such that $d(T, T') = 1$, all these trees can be enumerated in $|X|^{\mathcal{O}(1)}$ time, and for each binary X-tree T' with $d(T, T') > 0$, there is a binary X-tree $\hat{T}$ with $d(\hat{T}, T') = 1$ and $d(T, T') = d(T, \hat{T}) + 1$.

Furthermore, we may enumerate all binary X-trees T' with $d_{\text{sECR}}(T, T') \leq k$ as follows: First, we enumerate all subtrees of T with at most k edges. Second, for each connected subtree T_s of T with at most k edges, we enumerate all binary refinements of T after contracting all edges of T_s. In Lemma 5.2, we show that the first step can be done in $\mathcal{O}(4^k \cdot k^{-0.5} \cdot |X|)$ time. In Lemma 5.3, we show that both steps can be performed in $\mathcal{O}((2k+1)!! \cdot 4^k \cdot k\sqrt{k} \cdot |X|^2)$ time where $(2k+1)!! := 1 \cdot 3 \cdot \ldots \cdot (2k+1)$.

▶ **Lemma 5.2 (*).** *For every binary X-tree T and every integer k, all connected subtrees of T with at most k edges can be enumerated in $\mathcal{O}(4^k \cdot k^{-0.5} \cdot |X|)$ time.*

▶ **Lemma 5.3 (*).** *For a given binary X-tree T and an integer k, there are $\mathcal{O}((2k+1)!! \cdot 4^k \cdot k^{-0.5} \cdot |X|)$ binary X-trees T' with $d_{\text{sECR}}(T, T') \leq k$. Moreover, all these binary X-tree can be enumerated in $\mathcal{O}((2k+1)!! \cdot 4^k \cdot k\sqrt{k} \cdot |X|^2)$ time.*

Hence, we obtain the following due to the fact that the parsimony score of a given X-tree can be computed in $\mathcal{O}(|X| \cdot |S|)$ time [11].

▶ **Theorem 5.4.** *d_{sECR}-LS Maximum Parsimony can be solved in $\mathcal{O}((2k+1)!! \cdot 4^k \cdot k\sqrt{k} \cdot |X|^2 \cdot |S|) = 2^{\mathcal{O}(k \cdot \log k)} \cdot |X|^2 \cdot |S|$ time.*

Finally, we describe how to enumerate all binary X-trees T' with $d_{\text{ECR}}(T, T') \leq k$.

▶ **Lemma 5.5.** *Let T be a binary X-tree and let k be an integer. One can enumerate all binary X-trees T' with $d_{\text{ECR}}(T, T') \leq k$ in $|X|^{\mathcal{O}(k)}$ time.*

Proof. We show this statement by induction over k.

Base case. Consider $k = 0$. Hence, T is the only binary X-tree T' with $d_{\text{ECR}}(T, T') = 0$ and can be enumerated in $|X|^{\mathcal{O}(1)}$ time.

Inductive step. For the inductive step, suppose that for each binary X-tree $\tilde{T}$ and for each $k' < k$, one can compute all binary X-trees T' with $d_{\mathrm{ECR}}(\tilde{T}, T') \leq k'$ in $|X|^{\mathcal{O}(k')}$ time. Note that this implies that for each $k' < k$ there are $|X|^{\mathcal{O}(k')}$ binary X-trees T' with $d_{\mathrm{ECR}}(\tilde{T}, T') = k'$. For each $i < k$, let $\mathcal{T}_i$ be the collection of all binary X-trees $\tilde{T}$ with $d_{\mathrm{ECR}}(T, \tilde{T}) = i$ and let $\mathcal{T}_{<k}$ be the collection of all binary X-trees $\tilde{T}$ with $d_{\mathrm{ECR}}(T, \tilde{T}) < k$, that is, $\mathcal{T}_{<k} = \cup_{i=0}^{k-1} \mathcal{T}_i$. Moreover, let $\mathcal{T}_{\mathrm{sECR}}$ be the collection of all binary X-trees $\tilde{T}$ with $d_{\mathrm{sECR}}(T, \tilde{T}) = k$. Note that $\mathcal{T}_{<k}$ can be computed in $|X|^{\mathcal{O}(k-1)}$ time and due to Lemma 5.3, $\mathcal{T}_{\mathrm{sECR}}$ can be computed in $k^{\mathcal{O}(k)} \cdot |X|^{\mathcal{O}(1)}$ time. Let

$$\mathcal{T}'_k := \mathcal{T}_{\mathrm{sECR}} \cup \bigcup_{i=1}^{k-1} \bigcup_{\tilde{T} \in \mathcal{T}_i} \{T' \mid d_{\mathrm{ECR}}(\tilde{T}, T') \leq k - i\}.$$

Recall that by the induction hypothesis, for each $i < k$, $\mathcal{T}_i$ has size $|X|^{\mathcal{O}(i)}$ and for each binary X-tree $\tilde{T} \in \mathcal{T}_i$ the collection $\{T' \mid d_{\mathrm{ECR}}(\tilde{T}, T') \leq k - i\}$ can be computed in $|X|^{\mathcal{O}(k-i)}$ time. Hence, $\mathcal{T}'_k$ can be computed in $|X|^{\mathcal{O}(k)}$ time. We set $\mathcal{T} := \mathcal{T}'_k \cup \mathcal{T}_{<k}$ and show that $\mathcal{T}$ contains exactly the binary X-trees T' with $d_{\mathrm{ECR}}(T, T') \leq k$.

Assume towards a contradiction that this is not the case.

Case 1: There is a binary X-tree T' with $d_{\mathrm{ECR}}(T, T') \leq k$ such that T' is not in $\mathcal{T}$. By definition, $\mathcal{T}_{<k}$ contains all binary X-trees $\tilde{T}$ with $d_{\mathrm{ECR}}(T, \tilde{T}) < k$. Consequently, $d_{\mathrm{ECR}}(T, T') = k$. Hence, due to Observation 3.3, there is a binary X-tree $\tilde{T}$ with $d_{\mathrm{sECR}}(\tilde{T}, T') > 0$ such that $d_{\mathrm{ECR}}(T, T') = d_{\mathrm{ECR}}(T, \tilde{T}) + d_{\mathrm{sECR}}(\tilde{T}, T')$. Let $i := d_{\mathrm{ECR}}(T, \tilde{T})$.

Note that $i \leq k - 1$. If $i = 0$, then T is isomorphic to $\tilde{T}$ and thus $d_{\mathrm{sECR}}(T, T') = d_{\mathrm{sECR}}(\tilde{T}, T') = k$. Hence, T' is contained in $\mathcal{T}_{\mathrm{sECR}}$, a contradiction. Otherwise, if $i > 0$, then $\tilde{T}$ is contained in $\mathcal{T}_i$. Moreover, since $d_{\mathrm{sECR}}(\tilde{T}, T') = d_{\mathrm{ECR}}(T, T') - d_{\mathrm{ECR}}(T, \tilde{T}) = k - i$ and $d_{\mathrm{sECR}}(\tilde{T}, T') \geq d_{\mathrm{ECR}}(\tilde{T}, T')$, we have $d_{\mathrm{ECR}}(\tilde{T}, T') \leq k - i$ which implies that T' is contained in $\mathcal{T}$, a contradiction.

Case 2: There is a binary X-tree T' with $d_{\mathrm{ECR}}(T, T') > k$ such that T' is contained in $\mathcal{T}$. Hence, T' is contained in $\mathcal{T}'_k \setminus \mathcal{T}_{\mathrm{sECR}}$. That is, there is some i with $1 \leq i \leq k$ and a binary X-tree $\tilde{T}$ in $\mathcal{T}_i$ such that $d_{\mathrm{ECR}}(\tilde{T}, T') \leq k - i$. Since d_{ECR} is a metric, due to the triangle inequality, $d_{\mathrm{ECR}}(T, T') \leq d_{\mathrm{ECR}}(T, \tilde{T}) + d_{\mathrm{ECR}}(\tilde{T}, T') \leq k$, a contradiction.

Since $\mathcal{T}$ can be computed in $|X|^{\mathcal{O}(k)}$ time, the statement holds. ◀

We conclude the following.

▶ **Theorem 5.6 (*).** *For each distance measure $d \in \{d_{\mathrm{NNI}}, d_{\mathrm{ECR}}, d_{\mathrm{SPR}}, d_{\mathrm{TBR}}\}$, d-LS Maximum Parsimony can be solved in $|X|^{\mathcal{O}(k)} \cdot |S|$ time.*

6 Conclusion

A clear goal for future research would be to improve the running time of the algorithm for the k-sECR neighborhood. This seems promising since the current bottleneck is the enumeration of the binary refinements of the tree obtained after contracting k edges. However, an algorithm for d_{sECR}-LS Maximum Parsimony running in $2^{o(k \cdot \log k)} \cdot |I|^{\mathcal{O}(1)}$ time would imply an algorithm for Maximum Parsimony running in $2^{o(|X| \cdot \log |X|)} \cdot |I|^{\mathcal{O}(1)}$ time: when applying the d_{sECR}-LS Maximum Parsimony algorithm with $k := |X| - 3$, locally optimal solution are also globally optimal. Hence, a more immediate question is whether Maximum

Parsimony can be solved in $2^{o(|X| \cdot \log |X|)} \cdot |I|^{\mathcal{O}(1)}$ time. A further goal would be to find other neighborhoods for which d-LS Maximum Parsimony can be solved in time $f(k) \cdot |I|^{\mathcal{O}(1)}$. Finally, it is open whether better running times are possible when searching the neighborhood not for a better tree but for a perfect phylogeny, that is, for a tree where for each character, the parsimony score is equal to the number of character states minus one.

References

1 Benjamin L. Allen and Mike Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Ann. Comb.*, 5(1):1–15, 2001.

2 Alexandre A. Andreatta and Celso C. Ribeiro. Heuristics for the phylogeny problem. *J. Heuristics*, 8(4):429–447, 2002.

3 Hans L. Bodlaender, Michael R. Fellows, and Tandy J. Warnow. Two strikes against perfect phylogeny. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP '92)*, volume 623 of *Lecture Notes in Computer Science*, pages 273–283. Springer, 1992.

4 Amir Carmel, Noa Musa-Lempel, Dekel Tsur, and Michal Ziv-Ukelson. The worst case complexity of maximum parsimony. *J. Comput. Biol.*, 21(11):799–808, 2014.

5 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

7 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

8 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, and Yngve Villanger. Local search: Is brute-force avoidable? *J. Comput. Syst. Sci.*, 78(3):707–719, 2012. doi:10.1016/j.jcss.2011.10.003.

9 Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates Sunderland, 2004.

10 David Fernández-Baca and Jens Lagergren. A polynomial-time algorithm for near-perfect phylogeny. *SIAM J. Comput.*, 32(5):1115–1127, 2003. doi:10.1137/S0097539799350839.

11 Walter M. Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

12 Les R. Foulds and Ronald L. Graham. The Steiner problem in phylogeny is NP-complete. *Adv. Appl. Math.*, 3(1):43–49, 1982.

13 Ganeshkumar Ganapathy, Vijaya Ramachandran, and Tandy Warnow. On contract-and-refine transformations between phylogenetic trees. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, pages 900–909, 2004.

14 Ganeshkumar Ganapathy, Vijaya Ramachandran, and Tandy J. Warnow. Better hill-climbing searches for parsimony. In *Proceedings of the 3rd International Workshop on Algorithms in Bioinformatics (WABI '03)*, volume 2812 of *Lecture Notes in Computer Science*, pages 245–258. Springer, 2003. doi:10.1007/978-3-540-39763-2_19.

15 Serge Gaspers, Eun Jung Kim, Sebastian Ordyniak, Saket Saurabh, and Stefan Szeider. Don't be strict in local search! In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI '12)*. AAAI Press, 2012.

16 Adrien Goëffon, Jean-Michel Richer, and Jin-Kao Hao. Local search for the maximum parsimony problem. In *Proceedings of the First International Conference on Advances in Natural Computation (ICNC '05)*, volume 3612 of *Lecture Notes in Computer Science*, pages 678–683. Springer, 2005.

17 Adrien Goëffon, Jean-Michel Richer, and Jin-Kao Hao. Progressive tree neighborhood applied to the maximum parsimony problem. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 5(1):136–145, 2008. doi:10.1109/TCBB.2007.1065.

18 Pablo A Goloboff. Character optimization and calculation of tree lengths. *Cladistics*, 9(4):433–436, 1993.

19 Pablo A. Goloboff. Analyzing large data sets in reasonable times: Solutions for composite optima. *Cladistics*, 15(4):415–428, 1999. doi:10.1006/clad.1999.0122.

20 Maozu Guo, Jian-Fu Li, and Yang Liu. Improving the efficiency of p-ECR moves in evolutionary tree search methods based on maximum likelihood by neighbor joining. In *Proceeding of the Second International Multi-Symposium of Computer and Computational Sciences (IMSCCS '07)*, pages 60–67. IEEE Computer Society, 2007.

21 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

22 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

23 Christian Komusiewicz and Nils Morawietz. Parameterized local search for vertex cover: When only the search radius is crucial. In *Proceedings of the 17th International Symposium on Parameterized and Exact Computation (IPEC '22)*, volume 249 of *LIPIcs*, pages 20:1–20:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

24 Dániel Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.

25 Kevin C. Nixon. The parsimony ratchet, a new method for rapid parsimony analysis. *Cladistics*, 15(4):407–414, 1999. doi:10.1006/clad.1999.0121.

26 Celso C. Ribeiro and Dalessandro Soares Vianna. A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *Int. Trans. Oper. Res.*, 12(3):325–338, 2005.

27 David F. Robinson. Comparison of labeled trees with valency three. *J. Comb. Theory B*, 11(2):105–119, 1971.

28 David Sankoff. Minimal mutation trees of sequences. *SIAM J. Appl. Math.*, 28(1):35–42, 1975.

29 David Sankoff, Yvon Abel, and Jotun Hein. A tree · a window · a hill; generalization of nearest-neighbor interchange in phylogenetic optimization. *J. Classif.*, 11(2):209–232, 1994.

30 Srinath Sridhar, Kedar Dhamdhere, Guy E. Blelloch, Eran Halperin, R. Ravi, and Russell Schwartz. Algorithms for efficient near-perfect phylogenetic tree reconstruction in theory and practice. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 4(4):561–571, 2007. doi:10.1109/TCBB.2007.1070.