On the Weighted Quartet Consensus Problem[*]

Manuel Lafond[1] and Celine Scornavacca[2]

1   Department of Mathematics and Statistics, University of Ottawa, Ottawa,
    Canada
    mlafond2@uottawa.ca
2   Institut des Sciences de l'Evolution, Université Montpellier, CNRS, IRD,
    EPHE, Montpellier, France
    celine.scornavacca@umontpellier.fr

──── **Abstract** ────

In phylogenetics, the *consensus* problem consists in summarizing a set of phylogenetic trees that all classify the same set of species into a single tree. Several definitions of consensus exist in the literature; in this paper we focus on the Weighted Quartet Consensus problem, a problem with unknown complexity status so far. Here we prove that the Weighted Quartet Consensus problem is NP-hard and we give a 1/2-factor approximation for this problem. During the process, we propose a derandomization procedure of a previously known *randomized* 1/3-factor approximation. We also investigate the fixed-parameter tractability of this problem.

1   Introduction

Phylogenetics is the branch of biology that studies evolutionary relationships among different species. These relationships are represented via *phylogenetic trees* – acyclic connected graphs with leaves labeled by species – which are reconstructed from molecular and morphological data [12]. One fundamental problem in phylogenetics is to summarize the information contained in a set of unrooted trees $\mathcal{T}$ classifying the same set of species into a single tree T. The set $\mathcal{T}$ can consist of optimal trees for a single data set, of trees issued from a bootstrap analysis of a unique data set, or even of trees issued from the analysis of different data sets. Several consensus methods have been proposed in the past, the probably most known are the strict consensus [23, 18] and the majority-rule consensus [17, 3]. For a survey, see [7].

In this paper we focus on the Weighted Quartet Consensus (WQC) problem [19], also called the Median Tree with Respect to Quartet Distance problem [2] and Quartet Consensus problem in [16]. Roughly speaking, this problem consists in finding a consensus tree maximizing the weights of the 4-leaf trees – *quartets* – it displays, where the weight of a quartet is defined as its frequency in the set of input trees (for a more formal definition, see next section).

More general versions of this problem, where the input trees are allowed to have missing species or where the weight of a quartet is not defined w.r.t. a set of trees, are known to be

NP-hard [24] (and in fact, even Max-SNP-Hard), but the complexity of the WQC problem has remained open so far. This problem has been conjectured to be NP-hard [2, 19], and heuristics have recently been implemented and widely used, for instance ASTRAL [20], which is a practical implementation of Bryant and Steel's work from[8] (in fact, we show that the ASTRAL algorithm is a 2-approximation for the minimization version of WQC). So far, the best known approximation algorithm for the WQC problem consists in choosing a random tree as a solution [16]. This tree is expected to contain a third of the quartets from the input trees, and so it is a randomized factor 1/3 approximation. In [2], the *minimization* version of the problem is studied, where the objective is to find a median tree T minimizing the sum of quartet *distances* between T and the input trees (the quartet distance between two trees T_1 and T_2 is defined as the number of quartets in T_1 that are not in T_2). A 2-approximation algorithm is given, based on the fact that the quartet distance is a metric [9, 2].

A related problem that has received more attention is the *Complete Maximum Quartet Compatibility* problem (CMQC) (see [5, 4, 16, 14, 25, 26, 10, 21, 22]). In the CMQC problem, we are given, for each set S of four species, exactly one quartet on S, and the objective is to find a tree containing a maximum number of quartets from the input. This can be seen as a version of WQC in which each set of four species has one quartet of weight 1, and the others have weight 0. The CMQC and WQC are however fundamentally different. Although one could apply an algorithm for CMQC to WQC (by keeping only the most frequent quartet on each set of four taxa), maximizing the most-frequent quartets may enforce the presence of many low-frequency quartets. A better solution may prefer more of the middle-frequency quartets, and we give an example of this phenomenon. It was shown in [16] that the CMQC problem admits a polynomial time approximation scheme (PTAS), but it can only be extended to WQC intances on a constant number of trees. Also, CMQC was shown in [14, 10] to be fixed-parameter tractable w.r.t. the parameter "number of quartets to reject from the input".

The main result of this paper is to establish the NP-hardness of the WQC problem. In Section 2, we introduce preliminary notions, and in Section 3 we propose a reduction from the NP-hard CYCLIC ORDERING problem to WQC. It can be shown that this hardness result transfers to the *rooted* setting, in which case we want to optimize *triplets* (3-leaf rooted trees) rather than quartets. In Section 4, we discuss how being in a consensus setting, i.e. having weights based on a set of input trees on the same leaf set rather than arbitrary weights, does not necessarily make the problem easier, as one could expect: We list some structural properties that, surprisingly, are not satisfied by optimal solutions of a WQC instance. Nevertheless, in Section 5 we devise a factor 1/2 approximation algorithm for WQC running in time $O(k^2n^2 + kn^4 + n^5)$, where k is the number of trees and n the number of species (the best known randomized algorithm in the non-consensus setting is a factor 1/3 one). As a matter of fact, our algorithm includes a derandomization of this procedure, which had never been done before. Finally in Section 6, we show that the known FPT algorithms for the CMQC problem can be extended to the consensus setting. This yields an FPT algorithm that is efficient on instances in which there is not too much ambiguity, i.e. when few competing quartets on the same 4 species appear with the same frequency. We then conclude with some remarks and open problems related to the quartet consensus problem.

2   Preliminaries

An *unrooted phylogenetic tree T* consists of vertices connected by edges, in which every pair of nodes is connected by exactly one path and no vertex is of degree two. The *leaves* of a tree

T, denoted by $L(T)$ are the set of vertices of degree one. Each leaf is associated to a label; the set of labels associated to the leaves of a tree T is denoted by $\mathcal{L}(T)$. We suppose that there is a bijection between $L(T)$ and $\mathcal{L}(T)$. Due to this bijectivity, we will refer to leaves and labels interchangeably. We denote $|\mathcal{L}(T)|$ as $|T|$. In the following, we will often omit the word "phylogenetic" and, unless otherwise stated, all trees are leaf-labeled. A *binary* unrooted tree has only vertices with degree three and vertices with degree one. A rooted (binary) phylogenetic tree is defined in the same way, except that it has exactly one node of degree two called the *root*, denoted by $r(T)$. Note that sometimes in the literature, rooted trees are seen as directed and such that all arcs are oriented away from the root. Unless stated otherwise, all trees in this paper are unrooted.

Given an unrooted phylogenetic tree T and a subset $Y \subseteq \mathcal{L}(T)$, we denote by $T|Y$ the tree obtained from the minimal subgraph of T connecting Y when contracting degree-2 vertices. A *quadset* is a set of four labels. For a quadset $\{a, b, c, d\}$, there are three non-isomorphic[1] unrooted binary trees, called *quartets*, which are denoted respectively by $ab|cd$, $ac|bd$, and $ad|bc$, depending on how the central edge splits the four labels. We say that an unrooted tree T *displays* the quartet $ab|cd$ if $T|\{a, b, c, d\}$ is $ab|cd$. We denote the set of quartets that an unrooted tree T displays by $Q(T)$. Note that if T is binary, then $|Q(T)| = \binom{|\mathcal{L}(T)|}{4}$. A set of quartets Q on a set L is said to be *complete* if for each quadset $\{a, b, c, d\} \subseteq L$, there is in Q *exactly* one quartet among $ab|cd$, $ac|bd$, and $bc|ad$.

We are now ready to state our optimization problem. The WEIGHTED QUARTET CONSENSUS problem asks for a tree that has as many quartets as possible in common with a given set of trees on the same set of labels $\mathcal{X}$:

WEIGHTED QUARTET CONSENSUS (WQC) problem
Input: a set of unrooted trees $\mathcal{T} = \{T_1, \dots, T_k\}$ such that $\mathcal{L}(T_1) = \dots = \mathcal{L}(T_k) = \mathcal{X}$.
Output: a binary unrooted tree M with $\mathcal{L}(M) = \mathcal{X}$ that maximizes $\sum_{T \in \mathcal{T}} |Q(M) \cap Q(T)|$.

The problem is weighted as each quartet on $\mathcal{X}$ is weighted by frequency in $\mathcal{T}$, see below.

In this paper we will focus on the particular case where the input trees are all binary. In fact, proving the problem NP-hard for this restricted case implies NP-hardness of the general problem. Note however that relaxing the requirement of the output M to be binary leads to a different problem, as one needs to define how unresolved quartets in M are weighted.

In the remainder of the article, we will sometimes consider multi-sets of quartets, that are sets in which the same quartet can appear multiple times. Denote by $f_{\mathcal{Q}}(q)$ the number of times that a quartet q appears in a multi-set $\mathcal{Q}$ (we may write $f(q)$ if $\mathcal{Q}$ is unambiguous). We say that a tree T *contains* k quartets of $\mathcal{Q}$ if there are distinct quartets $q_1, \dots, q_p \in Q(T)$ such that $\sum_{i=1}^{p} f(q_i) = k$. The WEIGHTED QUARTET CONSENSUS problem can be rephrased as follows: given trees $T_1, \dots, T_k$, finding a tree M that contains a maximum number of quartets from $Q(T_1) \uplus Q(T_2) \uplus \dots \uplus Q(T_k)$, where $\uplus$ denotes multiset union. We will implicitly work with the decision version of WQC, i.e. for a given integer q, is there a consensus tree M containing at least q quartets from $Q(T_1) \uplus Q(T_2) \uplus \dots \uplus Q(T_k)$?

Given a quadset $\{a, b, c, d\}$, we say that $ab|cd$ is *dominant* (w.r.t. f) if $f(ab|cd) \geq f(ac|bd)$ and $f(ab|cd) \geq f(ad|bc)$. We say that $ab|cd$ is *strictly dominant* if both inequalities are strict.

---

[1] Isomorphism preserving labels.

3   NP-hardness of the Weighted Quartet Consensus problem

In this section, we present a reduction from the CYCLIC ORDERING problem. This problem has been used in phylogenetics before in [15] in the context of inferring rooted binary trees from rooted triplets that are not required to originate from a set of trees on the same leaf set.

But beforehand, we need some additional notation. A *caterpillar* is an unrooted binary tree obtained by taking a path $P = p_1 p_2 \ldots p_r$, then adding a leaf ℓ_i adjacent to p_i for each $1 \leq i \leq r$, then adding another leaf ℓ'_1 adjacent to p_1 and a leaf ℓ'_r adjacent to p_r. The two leaves ℓ'_1 and ℓ'_r inserted last are called the *ends* of the caterpillar. An *augmented caterpillar* T is a binary tree obtained by taking a caterpillar, then replacing each leaf by a binary rooted tree (replacing a leaf ℓ by a tree T' means replacing ℓ by $r(T')$). If T_1, T_2 are the two trees replacing the ends of the caterpillar, then T is called a (T_1, T_2)-augmented caterpillar. Note that every binary tree is a (T_1, T_2)-augmented caterpillar for some T_1, T_2. Let T be a caterpillar with leaves $(\ell_1, \ell_2, \ldots, \ell_k)$ taken in the order in which we encounter them on the path between the two ends l_1 and l_k (more precisely, traverse the $\ell_1 - \ell_k$ path, and each time an internal node is encountered, append its adjacent leaves to the sequence), and let $T_1, \ldots, T_k$ be rooted binary trees. We denote by $(T_1 | T_2 | \ldots | T_k)$ the (T_1, T_k)-augmented caterpillar obtained by replacing ℓ_i by $r(T_i)$, $1 \leq i \leq k$. This notation gives rise to a natural ordering of its subtrees, and we say that $T_i < T_j$ if $i < j$ (i.e. T_i appears before T_j in the given ordering of the caterpillar subtrees). Note that by reversing such an ordering, we obtain the same unrooted tree. However, in the proofs the given ordering will be important. Also, since T_1, T_2, and T_{k-1}, T_k are interchangeable, we will simply say that these two pairs are incomparable. If each T_i consists of a single leaf ℓ_i for $2 \leq i \leq k - 1$, then we may write $(T_1 | \ell_2 | \ldots | \ell_{k-1} | T_k)$, and $\ell_i < \ell_j$ in T to indicate that ℓ_i appears before ℓ_j in the ordering.

We are now ready to describe the CYCLIC ORDERING problem. Let $L = (s_1, \ldots, s_n)$ be a linear ordering of a set S of at least 3 elements, and let (a, b, c) be an ordered triple, with $a, b, c \in S$. We say that L *satisfies* (a, b, c) if one of the following holds in L: $a < b < c, b < c < a$ or $c < a < b$. If $\mathcal{C}$ is a set of ordered triples we say that L satisfies $\mathcal{C}$ if it satisfies every element of $\mathcal{C}$. Intuitively speaking, L satisfies (a, b, c) when, by turning L into a directed cyclic order by attaching s_n to s_1, one can go from a to b, then to c and then to a. This leads to the following problem definition:

CYCLIC ORDERING problem
Input: A set S of n elements and a set $\mathcal{C}$ of m ordered triples (a, b, c) of distinct members of S.
Question: Does there exist a linear ordering $L = (s_1, \ldots, s_n)$ of S that satisfies $\mathcal{C}$?

The CYCLIC ORDERING problem is NP-hard [13]. In the rest of this section, we let S and $\mathcal{C}$ be the input set and triples, respectively, of a CYCLIC ORDERING problem instance. We denote $n = |S|$ and $m = |\mathcal{C}|$. We shall use the following simple yet useful characterization.

▶ **Lemma 1.** *A linear ordering L of S satisfies $\mathcal{C}$ if and only if for each $(a, b, c) \in \mathcal{C}$, exactly two of the following relations hold in L: $a < b, b < c, c < a$.*

Proof. ($\Rightarrow$): let L be a linear ordering satisfying $\mathcal{C}$, and let $(a, b, c) \in \mathcal{C}$. Thus in L, one of $a < b < c, b < c < a$ or $c < a < b$ holds. It is straightforward to verify that in each case, exactly two of $a < b, b < c, c < a$ hold.
($\Leftarrow$): suppose that L does not satisfy $\mathcal{C}$. Then there is some $(a, b, c) \in \mathcal{C}$ such that one of $a < c < b, b < a < c$ or $c < b < a$ does not hold. Again, one can easily verify that each of these cases satisfies only one of $a < b, b < c$ and $c < a$. ◀

Now, from S and $\mathcal{C}$ we construct a set of unrooted binary trees $\mathcal{T}$ on the same set of labels (we will omit the straightforward verification that this construction can be carried out in polynomial time). First let W and Z be two rooted binary trees each on $(nm)^{100}$ leaves (the topology is arbitrary, and the 100 exponent could be optimized). Our trees are defined on the leaf set $\mathcal{X} = S \cup \mathcal{L}(W) \cup \mathcal{L}(Z)$ (note that $S, \mathcal{L}(W), \mathcal{L}(Z)$ are disjoint). Let $C \in \mathcal{C}$ with $C = (a, b, c)$. We construct 6 trees from C, that is, 3 pairs of trees:

- The "$a < b$" trees: let $(s_1, \ldots, s_{n-2})$ be an arbitrary ordering of $S \setminus \{a, b\}$. Then we build the trees $T_C(ab) = (W|a|b|s_1|s_2|\ldots|s_{n-2}|Z)$ and $\overleftarrow{T}_C(ab) = (W|s_{n-2}|s_{n-3}|\ldots|s_1|a|b|Z)$.
- The "$b < c$" trees: let $(\hat{s}_1, \ldots, \hat{s}_{n-2})$ be an arbitrary ordering of $S \setminus \{b, c\}$. Then we build the trees $T_C(bc) = (W|b|c|\hat{s}_1|\hat{s}_2|\ldots|\hat{s}_{n-2}|Z)$ and $\overleftarrow{T}_C(bc) = (W|\hat{s}_{n-2}|\hat{s}_{n-3}|\ldots|\hat{s}_1|b|c|Z)$.
- The "$c < a$" trees: let $(\bar{s}_1, \ldots, \bar{s}_{n-2})$ be an arbitrary ordering of $S \setminus \{c, a\}$. Then we build the trees $T_C(ca) = (W|c|a|\bar{s}_1|\bar{s}_2|\ldots|\bar{s}_{n-2}|Z)$ and $\overleftarrow{T}_C(ca) = (W|\bar{s}_{n-2}|\bar{s}_{n-3}|\ldots|\bar{s}_1|c|a|Z)$.

Denote by $\mathcal{T}(C)$ the set of 6 constructed trees for $C \in \mathcal{C}$. In this section, the input for our WEIGHTED QUARTET CONSENSUS instance constructed from S and $\mathcal{C}$ is $\mathcal{T} = \bigcup_{C \in \mathcal{C}} \mathcal{T}(C)$. Note that $|\mathcal{T}| = 6m$.

Observe that each tree of $\mathcal{T}(C)$ is a (W, Z)-augmented caterpillar. Moreover, note that the majority of ordered pairs are "balanced" in the pairs of constructed trees: Let $a, b \in S$ and $x, y \in S \setminus \{a, b\}$, and let $\{T_C(ab), \overleftarrow{T}_C(ab)\}$ be an "$a < b$" tree-pair. Then we have $x < y$ in $T_C(ab)$ if and only if $y < x$ in $\overleftarrow{T}_C(ab)$. Similarly for any $x \in S \setminus \{a, b\}$, $a < x, b < x$ in $T_C(ab)$ but $x < a, x < b$ in $\overleftarrow{T}_C(ab)$. Only $a < b$ holds in both trees.

Let $T \in \mathcal{T}$, and let $B(T)$ denote the set of quartets of T that have at least two members of $\mathcal{L}(W)$, or at least two members of $\mathcal{L}(Z)$. Thus $B(T)$ consists in all the quartets of the form $w_1 w_2|xy$ and $z_1 z_2|xy$ of T, where $w_1, w_2 \in \mathcal{L}(W), z_1, z_2 \in \mathcal{L}(Z)$ and $x, y \in \mathcal{X}$ (note that no quartet of $B(T)$ has the form $w_1 x|y w_2$ for $x, y \notin \mathcal{L}(W)$, nor the form $z_1 x|y z_2$ for $x, y \notin \mathcal{L}(Z)$). Note that for any tree $T' \in \mathcal{T}$, $B(T) = B(T')$. Let $K := 6m|B(T)|$ be the total number of such quartets in $\mathcal{T}$, i.e. K is the size of $\biguplus_{T \in \mathcal{T}} B(T)$. We observe the following:

▶ **Remark.** Any (W, Z)-augmented caterpillar on $\mathcal{X}$ contains the K quartets $\biguplus_{T \in \mathcal{T}} B(T)$.

Now, denote $\hat{O} := 3m|W||Z| \left(\binom{n-2}{2} + 2(n-2) \right)$. Let $T \in \mathcal{T}$ and suppose that T is an "$a < b$" tree, for some $a, b \in S$. For $w \in \mathcal{L}(W)$ and $z \in \mathcal{L}(Z)$, $x, y \in S$, a quartet $wx|yz$ displayed by T is called an *out-quartet* of T if $\{x, y\} \neq \{a, b\}$, and an *in-quartet* of T if $x = a$ and $y = b$ (note that $x = b$ and $y = a$ is not possible, by construction). Let $out(T)$ and $in(T)$ denote the set of out-quartets and in-quartets, respectively, of T. Note that each tree T has $|W||Z|$ in-quartets and $|W||Z| \left(\binom{n-2}{2} + 2(n-2) \right)$ out-quartets (because there are $\binom{n-2}{2} + 2(n-2)$ ways to choose $\{x, y\} \neq \{a, b\}$). Thus $\hat{O}$ is half the total number of out-quartets.

▶ **Lemma 2.** *Any weighted quartet consensus tree M for $\mathcal{T}$ contains at most $\hat{O}$ quartets from $\biguplus_{T \in \mathcal{T}} out(T)$. Moreover, if M is a (W, Z)-augmented caterpillar $(W|s_1|\ldots|s_n|Z)$, where $S = \{s_1, \ldots, s_n\}$, then M contains exactly $\hat{O}$ quartets from $\biguplus_{T \in \mathcal{T}} out(T)$.*

Proof. Let $w \in \mathcal{L}(W)$ and $z \in \mathcal{L}(Z)$. Let $\{T_C(ab), \overleftarrow{T}_C(ab)\}$ be an "$a < b$" tree-pair of $\mathcal{T}$, for some $a, b \in S$, and let $x, y \in S$ such that $\{x, y\} \neq \{a, b\}$. Because $x < y$ in $T_C(ab)$ if and only if $y < x$ in $\overleftarrow{T}_C(ab)$, we get that the out-quartet $wx|yz$ is in $T_C(ab)$ if and only if $wy|xz$ is in $\overleftarrow{T}_C(ab)$. Since M can only contain one of the two quartets, it follows that M can contain at most half of the quartets from $out(T_C(ab)) \uplus out(\overleftarrow{T}_C(ab))$. Thus M contains at most half the quartets from $\biguplus_{T \in \mathcal{T}} out(T)$, which is $3m|W||Z| \left(\binom{n-2}{2} + 2(n-2) \right) = \hat{O}$. As for the second assertion, if $M = (W|s_1|\ldots|s_n|Z)$ then M contains one of $wx|yz$ or $wy|xz$ for each $x, y \in S$. Thus if M does not contain the out-quartet $wx|yz$ from $T_C(ab)$, then it

contains the out-quartet $wy|xz$ from $\overleftarrow{T}_C(ab)$. We deduce that M contains at least half the quartets from $out(T_C(ab)) \uplus out(\overleftarrow{T}_C(ab))$, and thus half the quartets from $\uplus_{T \in \mathcal{T}} out(T)$. ◄

What follows is a key Lemma. The proof is not so straightforward and can be found in Appendix B.1.

▶ **Lemma 3.** *Any optimal consensus tree for $\mathcal{T}$ is a (W, Z)-augmented caterpillar.*

We finally arrive at our main result.

▶ **Theorem 4.** *The* WEIGHTED QUARTET CONSENSUS *problem is NP-hard.*

Proof. We show that there exists a linear ordering of S satisfying $\mathcal{C}$ if and only if there exists a weighted quartet consensus tree M for $\mathcal{T}$ that contains at least $K + \hat{O} + 4m|W||Z|$ quartets from $\uplus_{T \in \mathcal{T}} Q(T)$. For the rest of the proof, we let $w \in \mathcal{L}(W)$ and $z \in \mathcal{L}(Z)$ be arbitrary leaves of W and Z, respectively.

($\Rightarrow$): let $L = (s_1, s_2, \ldots, s_n)$ be a linear ordering of S satisfying $\mathcal{C}$. Then we claim that the weighted quartet consensus tree $M = (W|s_1|s_2|\ldots|s_n|Z)$ contains the desired number of quartets. Since M is a (W, Z)-augmented caterpillar, M contains K quartets of $\mathcal{T}$ that have two or more elements from $\mathcal{L}(W)$, or two or more elements from $\mathcal{L}(Z)$, see remark on page 5. Moreover by Lemma 2, M contains $\hat{O}$ quartets from $\uplus_{T \in \mathcal{T}} out(T)$. As for the in-quartets, let $(a, b, c) \in \mathcal{C}$ and let $\mathcal{T}((a, b, c))$ be the set of 6 trees corresponding to (a, b, c). By Lemma 1, L satisfies two of the relations $a < b, b < c, c < a$. This implies that M has exactly two of the following quartets: $wa|bz, wb|cz, wc|az$. Since, for every $w \in \mathcal{L}(W)$ and $z \in \mathcal{L}(Z)$, each of these three quartets appears as an in-quartet in exactly two trees of $\mathcal{T}((a, b, c))$ (e.g. $wa|bz$ is an in-quartet of $T_{(a,b,c)}(ab)$ and $\overleftarrow{T}_{(a,b,c)}(ab)$), it follows that M contains $4|W||Z|$ quartets of $\uplus_{T \in \mathcal{T}((a,b,c))} in(T)$. As this holds for every $(a, b, c) \in C$, M contains $4m|W||Z|$ quartets of $\uplus_{T \in \mathcal{T}} in(T)$. Summing up, we get that M has at least $K + \hat{O} + 4m|W||Z|$ quartets from $\mathcal{T}$.

($\Leftarrow$): suppose that no linear ordering of S satisfies $\mathcal{C}$. Let M be an optimal consensus tree for $\mathcal{T}$. By Lemma 3, we may assume that M is a (W, Z)-augmented caterpillar. We bound the number of quartets of $\mathcal{T}$ that can be contained in M.

First, by Lemma 3, M contains K quartets of $\mathcal{T}$ that have at least two elements of $\mathcal{L}(W)$ or at least two elements of $\mathcal{L}(Z)$. As for the quartets with one or zero elements from $\mathcal{L}(W) \cup \mathcal{L}(Z)$, in any tree $T \in \mathcal{T}$ there are at most $(|W| + |Z|)n^3$ quartets of the form $wa|bc$ or $za|bc$ with $a, b, c \in S$, and at most n^4 quartets of the form $ab|cd$ with $a, b, c, d \in S$. Thus M contains at most $6m((|W| + |Z|)n^3 + n^4) < (|W| + |Z|)mn^5$ quartets of $\mathcal{T}$ that are of the form $wa|bc, za|bc$ or $ab|cd$ with $a, b, c \in S$ (the inequality holds because $n \geq 3$ and $|W| = |Z| = (nm)^{100}$). Also, by Lemma 2, M contains at most $\hat{O}$ quartets from $\uplus_{T \in \mathcal{T}} out(T)$. It remains to count the in-quartets.

Let $(a, b, c) \in \mathcal{C}$. The following in-quartets appear, each twice, in $\mathcal{T}((a, b, c))$: $wa|bz$, $wb|cz$, $wc|az$. It is easy to check that these three quartets are incompatible, i.e. no tree can contain all three of them, and hence M can have at most two of them. We deduce that there must be at least two trees $T, \overleftarrow{T}$ of $\mathcal{T}((a, b, c))$ such that M contains no quartet from $in(T) \uplus in(\overleftarrow{T})$. Therefore M contains at most $4|W||Z|$ quartets from $\uplus_{T \in \mathcal{T}((a,b,c))} in(T)$, and thus at most $4m|W||Z|$ quartets from $\uplus_{T \in \mathcal{T}} in(T)$ assuming that the $4|W||Z|$ bound is attained for every $(a, b, c) \in \mathcal{C}$. We will however show that there must be some $(a, b, c) \in \mathcal{C}$ such that M contains only $2|W||Z|$ of the quartets in $\uplus_{T \in \mathcal{T}((a,b,c))} in(T)$.

Now, since M is a (W, Z)-augmented caterpillar, we write $M = (W|T_1|T_2|\ldots|T_k|Z)$. For some $a \in S$, let $T(a)$ be the tree of $\{T_1, \ldots, T_k\}$ that contains a as a leaf. Then a quartet $wa|bz$ is in $Q(M)$ if and only if $T(a) < T(b)$. Let L be a linear ordering of S such

that $T(a) < T(b) \Rightarrow a < b$ in L. Since no linear ordering of S can satisfy $\mathcal{C}$, by Lemma 1 there must be some $(a, b, c) \in \mathcal{C}$ such that only one of $a < b, b < c, c < a$ holds in L. This also means that at most one of $T(a) < T(b), T(b) < T(c), T(c) < T(a)$ holds (because $\neg(a < b) \Rightarrow \neg(T(a) < T(b))$). Thus M has at most one of the $wa|bz, wb|cz, wc|az$ quartets. It follows that there are at least $2|W||Z|$ quartets from $\biguplus_{T \in \mathcal{T}((a,b,c))} in(T)$ that M does not contain. Therefore M contains at most $4m|W||Z| - 2|W||Z|$ quartets of $\biguplus_{T \in \mathcal{T}} in(T)$.

In total, the number of quartets that M contains from the input is bounded by $K + \hat{O} + (|W| + |Z|)mn^5 + (4m-2)|W||Z| < K + \hat{O} + 4m|W||Z|$, by our choice of $|W|$ and $|Z|$. ◄

The implications of these results for the Weighted Triplet Consensus (WTC) problem are presented in Appendix A. The same techniques can be used to show that WTC is NP-hard.

4   The (non)-structure of WQC

In the rest of this paper, we aim at designing algorithms building on the fact that the weight of each quartet is not arbitrary, and is rather based on a set of input trees on the same leaf set. When designing optimized algorithms for a problem, understanding the relationship between the input and the optimal solution(s) can be of great help. In phylogenetics, several problems are harder in the supertree setting, i.e. when the input trees do not all contain the same species, than in the consensus setting as in the WQC problem. An example is the problem of finding an unrooted phylogenetic tree containing as minors a set of unrooted phylogenetic trees – the compatibility problem – which is NP-hard in the supertree setting [24] and polynomially solvable in the consensus setting [1]. Despite the NP-hardness of WQC, there may exist some properties inherent to the consensus setting that are useful for devising efficient FPT algorithm, or for establishing lower bounds on the value of an optimal solution in order to develop approximation algorithms.

In attempt to establish useful properties of the weights of quartets in the consensus setting, we initially conjectured that the following relationships between the input trees and the solution(s) hold. Despite being seemingly reasonable, we prove all these conjectures false.

1. let D be the set of strictly dominant quartets of the input multiset $\mathcal{Q}$, i.e. the quartets $ab|cd$ such that $f(ab|cd) > f(ac|bd)$ and $f(ab|cd) > f(ad|bc)$. Then there is a constant $\alpha > 0$ such that there exists an optimal solution containing at least $\alpha|D|$ quartets of D;
2. if a quartet $ab|cd$ has a higher weight than the sum of the other quartets on the same quadset, i.e. $f(ab|cd) > f(ac|bd) + f(ad|bc)$, then some optimal solution contains $ab|cd$;
3. more generally, there exists $\beta > 0$ such that if a quartet $ab|cd$ is in a fraction β of the input trees, then $ab|cd$ must be in some optimal solution. In particular, if $ab|cd$ is in *every* input tree, then there is some optimal solution that contains $ab|cd$;
4. if a quartet $ab|cd$ is in no input tree, then no optimal solution contains $ab|cd$.
5. call $ab|cd$ a *strictly least-frequent quartet* if $f(ab|cd) < f(ac|bd)$ and $f(ab|cd) < f(ad|bc)$. Suppose that there exists a tree T^* on leaf set $\mathcal{X}$ that contains no strictly least-frequent quartet, and choose such a T^* that contains a maximum number of quartets from the input. Then T^* is an optimal solution for WQC.

Unfortunately, we answered negatively to all conjectures, see Appendix B.2.

5   Approximability of WQC

In this section, we show that WQC admits a factor $1/2$ approximation algorithm that runs in polynomial time. Hereafter, the input set of trees is $\mathcal{T} = \{T_1, \dots, T_k\}$ and we denote

$\mathcal{Q} = Q(T_1) \uplus \ldots \uplus Q(T_k)$. We say that a minimization (resp. maximization) problem P can be approximated within a factor $\alpha > 1$ (resp. $\beta < 1$) if there is an algorithm that, for every instance I of P, runs in polynomial time and outputs a solution of value $APP(I)$ such that $APP(I) \leq \alpha OPT(I)$ (resp. $APP(I) \geq \beta OPT(I)$), where $OPT(I)$ is the optimal value of I.

As mentioned before, the Complete Maximum Quartet Compatibility (CMQC) problem admits a PTAS, though it can only be applied to the WQC problem when the number of input trees is constant. There does not seem to exist an easy extension of the PTAS algorithm for the case of an unbounded number of trees, which makes WQC seem "harder" than CMQC. Nevertheless, we give a simple factor $1/2$ approximation algorithm, which is better than the (randomized) factor $1/3$ approximation, the best known so far, for the general Maximum Quartet Consistency problem in which the given quartet set is not necessarily complete. We borrow ideas from [9] to show that this can be achieved by taking the best solution from either a $1/3$ approximation to WQC, or a factor 2 approximation to WMQI, the minimization version of WQC (see below). For two unrooted binary trees T_1, T_2 on leaf set $\mathcal{X}$, denote $d_Q(T_1, T_2) = |Q(T_1) \setminus Q(T_2)|$. The WMQI problem is defined as follows:

WEIGHTED MINIMUM QUARTET INCONSISTENCY (WMQI) problem
Input: a set of unrooted trees $\mathcal{T} = \{T_1, \ldots, T_k\}$ such that $\mathcal{L}(T_1) = \ldots = \mathcal{L}(T_k) = \mathcal{X}$.
Output: a tree M with $\mathcal{L}(M) = \mathcal{X}$ that minimizes $\sum_{T \in \mathcal{T}} d_Q(M, T)$.

Note that the WMQI problem is equivalent to finding a minimum (in the multiset sense) number of quartets to discard from $\mathcal{Q}$ so that it is compatible.

It is not hard to show that d_Q is a metric. In particular, d_Q satisfies the triangle inequality, i.e. for any 3 trees T_1, T_2, T_3 on the same leaf set, $d_Q(T_1, T_3) \leq d_Q(T_1, T_2) + d_Q(T_2, T_3)$. This leads to a factor 2 approximation algorithm for WMQI obtained by simply returning the best tree from the input. Intuitively, the input tree that is the closest to the others cannot be too far from the best solution, which is a median tree in the metric space. See [2] for details.

▶ **Theorem 5** ([2]). *The following is a factor 2 approximation algorithm for WMQI: output the tree $T \in \mathcal{T}$ that minimizes $\sum_{T_i \in \mathcal{T}} d_Q(T, T_i)$.*

In [2], the authors explain how to compute $d_Q(T_1, T_2)$ in time $O(n^2)$. Therefore the factor 2 approximation can be implemented to run in time $O(k^2 n^2)$, by simply computing d_Q between every pair of trees.

Theorem 5 has a direct implication on the approximation guarantees of the ASTRAL algorithm in [20], an implementation of the work from Bryant and Steel [8]. This algorithm finds, in polynomial time, an optimal solution M for a restricted version of WMQI where every bipartition of M is also a bipartition in at least one of the input trees. The solution T returned by the algorithm of Theorem 5 above trivially satisfies this condition. Thus, M is at least as good as T, implying the following.

▶ **Corollary 6.** *The ASTRAL algorithm is a factor 2 approximation for WMQI.*

We do not know whether the factor 2 is tight for the ASTRAL algorithm - we conjecture that ASTRAL can actually achieve a better approximation ratio. As shown in the rest of this section, this would have interesting applications for the approximability of WQC.

Indeed, both WQC and WMQI share the same set of optimal solutions, but the two problems are not necessarily identical in terms of approximability. We show however that WMQI can be used to approximate WQC. As stated earlier, there is a trivial factor $1/3$ randomized approximation for WQC: output a random tree T. Each quartet of $\mathcal{Q}$ has a $1/3$ chance of being contained by T, and so the expected number of quartets of $\mathcal{Q}$ contained by

T is $|\mathcal{Q}|/3 = k\binom{n}{4}/3$ (here $|\mathcal{Q}|$ denotes the multiset cardinality). Call this the "random-tree-algorithm". For the sake of having a *deterministic* algorithm, we show the following:

▶ **Lemma 7.** *The "random-tree-algorithm" can be derandomized, i.e. there is a deterministic algorithm that, in time $O(kn^4 + n^5)$, finds a tree containing at least $|\mathcal{Q}|/3$ quartets from $\mathcal{Q}$.*

Proof. We derandomize the factor $1/3$ algorithm using the standard method of conditional expectation. For the simplicity of exposition, we will construct a rooted tree T in a top-down manner (T can be unrooted after the construction). Call a rooted tree T *internally binary* if the only nodes of T that have more than two children have only leaves as children. We start with a fully unresolved internally binary tree T on leaf set $\mathcal{X}$ (i.e. T consists of a root whose n children are in bijection with $\mathcal{X}$). We then iteratively split each unresolved node v of T into two subtrees so as to maximize the expected number of quartets that T contains. We stop when T is a binary tree.

To describe the algorithm more precisely, suppose that T is an internally binary tree on leaf set $\mathcal{X}$, and let v be a node of T with more than 2 children, say $\{v_1, \ldots, v_m\} \subseteq \mathcal{X}$ (if no such v exists, then T is binary and we can stop). We split v by first removing $\{v_1, \ldots, v_m\}$ from T, adding two children x and y to v, and reinserting $v_1, \ldots, v_m$ one after another, each as either a child of x or a child of y. We describe how this choice is made. Suppose that for $i \geq 1$, $v_1, \ldots, v_{i-1}$ have been reinserted, resulting in the tree T_{i-1}, and that we need to process v_i. Denote by $T_{i,x}$ (resp. $T_{i,y}$) the tree obtained by inserting v_i as a child of x (resp. of y) in T_{i-1}. We then define a random binary tree $T'_{i,x}$ from $T_{i,x}$ as follows: for each $v' \in \{v_{i+1}, \ldots, v_m\}$, reinsert v' as a child of x with probability $1/2$, or as a child of y with probability $1/2$. Then, replace each non-binary node w with children $\mathcal{X}'$ by a rooted binary tree on leaf set $\mathcal{X}'$ chosen uniformly at random. We define the random binary tree $T'_{i,y}$ from $T_{i,y}$ using the same process.

For a random tree T' obtained by the above process and for $q \in \mathcal{Q}$, let $I(q, T')$ be an indicator variable for whether $q \in Q(T')$. That is, $I(q, T') = 1$ if $q \in Q(T')$, and $I(q, T') = 0$ otherwise. Let $I(T') = \sum_{q \in \mathcal{Q}} I(q, T') f_{\mathcal{Q}}(q)$ [2]. We seek

$$\max_{T' \in \{T'_{i,x}, T'_{i,y}\}} \mathbb{E}\left[I(T')\right] = \max_{T' \in \{T'_{i,x}, T'_{i,y}\}} \mathbb{E}\left[\sum_{q \in \mathcal{Q}} I(q, T') f_{\mathcal{Q}}(q)\right]$$

$$= \max_{T' \in \{T'_{i,x}, T'_{i,y}\}} \sum_{q \in \mathcal{Q}} \Pr\left[q \in Q(T')\right] f_{\mathcal{Q}}(q).$$

If $T'_{i,x}$ attains this maximum, we insert v_i below x, and otherwise we insert v_i below y. After every child v_i of v has been inserted, we process the next non-binary node. This concludes the algorithm description (we shall detail how to compute $\Pr[q \in Q(T')]$ below).

If T is an internally binary tree, by a slight abuse of notation define $\mathbb{E}\left[I(T)\right] = \mathbb{E}\left[I(T')\right]$, where T' is the random binary tree obtained by replacing each non-binary node of T on leaf set $\mathcal{X}'$ by a random binary tree on leaf set $\mathcal{X}'$.

▶ **Claim 1.** *Let T be an internally binary tree, and suppose that $\mathbb{E}[I(T)] \geq |\mathcal{Q}|/3$. Let v be a non-binary node of T, and let T_v be the tree obtained after splitting v using the above algorithm. Then $\mathbb{E}[I(T_v)] \geq |\mathcal{Q}|/3$.*

---

[2] Observe that here, $q \in \mathcal{Q}$ means that there exists at least one occurrence of q in the multiset $\mathcal{Q}$, and so each quartet present in $\mathcal{Q}$ is considered once in the summation, independently of $f_{\mathcal{Q}}(q)$.

Let $\{v_1, \ldots, v_m\}$ be the children of v. To prove the claim, we use induction on the number of processed children of v to show that after each insertion of a child v_i, the obtained tree $T_i \in \{T_{i,x}, T_{i,y}\}$ satisfies $\mathbb{E}[I(T_i')] \geq |\mathcal{Q}|/3$, where $T_i' \in \{T_{i,x}', T_{i,y}'\}$ is the random tree corresponding to T_i obtained from the above process (i.e. reinserting $v_{i+1}, \ldots, v_m$ randomly under x or y, and resolving non-binary nodes randomly). This proves the statement since $T_m = T_v$ (and thus $\mathbb{E}[I(T_v)] = \mathbb{E}[I(T_m)] = \mathbb{E}[I(T_m')] \geq |\mathcal{Q}|/3$). As a base case, if $i = 1$ it is easy to see that $T_{1,x}'$ and $T_{1,y}'$ are identical, and that $\mathbb{E}[I(T_{1,x}')] = \mathbb{E}[I(T_{1,y}')] = \mathbb{E}[I(T)] \geq |\mathcal{Q}|/3$. For $i > 1$, let T_{i-1} be the tree obtained after inserting v_{i-1}, and suppose without loss of generality that $T_{i-1} = T_{i-1,x}$. Because, in $T_{i-1,x}'$, we insert v_i below x or y each with probability $\frac{1}{2}$, we have

$$\mathbb{E}\left[I(T_{i-1,x}')\right] = \frac{1}{2}\mathbb{E}\left[I(T_{i-1,x}')|v_i \text{ is a child of } x\right] + \frac{1}{2}\mathbb{E}\left[I(T_{i-1,x}')|v_i \text{ is a child of } y\right]$$

$$= \frac{1}{2}\left(\mathbb{E}\left[I(T_{i,x}')\right] + \mathbb{E}\left[I(T_{i,y}')\right]\right).$$

By induction, we also have $\mathbb{E}[I(T_{i-1,x}')] \geq |\mathcal{Q}|/3$. Combined with the above equality, we obtain $\frac{1}{2}\left(\mathbb{E}\left[I(T_{i,x}')\right] + \mathbb{E}\left[I(T_{i,y}')\right]\right) \geq |\mathcal{Q}|/3$. This implies that one of $\mathbb{E}[I(T_{i,x}')]$ or $\mathbb{E}[I(T_{i,y}')]$ must be at least $|\mathcal{Q}|/3$. ◄

Since the fully unresolved tree T from which we start satisfies $\mathbb{E}[I(T)] \geq |\mathcal{Q}|/3$, Claim 1 shows that the algorithm does terminate with a tree containing at least $|\mathcal{Q}|/3$ quartets from $\mathcal{Q}$. It remains to be show how to compute, when reinserting a node v_i, the expectations for $T_{i,x}'$ and $T_{i,y}'$.

In fact, it suffices to be able to compute, for a given quartet $q = ab|cd$, the probability $\Pr[q \in Q(T')]$ for $T' \in \{T_{i,x}', T_{i,y}'\}$. Moreover, if $\Pr[q \in Q(T_{i,x}')] = \Pr[q \in Q(T_{i,y}')]$, then this probability does not contribute to determining which scenario maximizes expectation, and in this case we do not need to consider q. In particular, if none of a, b, c, d is equal to v_i, then $\Pr[q \in Q(T_{i,x}')] = \Pr[q \in Q(T_{i,y}')]$. Therefore, it is enough to consider only quartets in which v_i is included. We will assume that $v_i = a$. Moreover, we may assume that two or three of $\{b, c, d\}$ are children of v in T (recall that v is the parent of v_i in T), because otherwise the probability that $ab|cd$ is in T' is unaffected by whether a is a child of x or a child of y.

There are still multiple cases depending on which of b, c and d are children of v, and which have been reinserted or have not, but this probability can be easily found algorithmically. Let $U = \{b, c, d\} \cap \{v_{i+1}, \ldots, v_m\}$, i.e. the leaves in $\{b, c, d\}$ that have not been reinserted yet in T'. We obtain new trees $S_1', \ldots, S_h'$ by reinserting, in T', the members of U below x or y in every possible way – there are only $2^{|U|} \leq 8$ possibilities, so $h \leq 8$. Then, for $1 \leq j \leq h$ denote by $S_j'|_q$ the tree S_j' restricted to $\{a, b, c, d\}$ (i.e. obtained by removing every leaf not in $\{a, b, c, d\}$, then contracting degree 2 vertices). Note that $S_j'|_q$ may be non-binary. We get $\Pr[q \in Q(T')] = \sum_{j=1}^h \frac{1}{h}\Pr[q \in Q(S_j'|_q)]$. This is because every leaf in $v_{i+1}, \ldots, v_m$ other than b, c, d is reinserted independently from the choice for b, c, d, and every non-binary node remaining after the reinsertions is resolved uniformly. The probability $\Pr[q \in Q(S_j'|_q)]$ is straightforward to compute, as only a constant number of cases can occur since $S_j'|_q$ has only 4 leaves. We omit the details.

Time complexity: we must first preprocess the input in order to compute $f_{\mathcal{Q}}(q)$ for each quartet q. This takes time $O(kn^4)$. As for the computation of $\Pr[q \in Q(T')]$, assume that the lowest common ancestor (lca) of two leaves can be found in constant time. This can be achieved naively by simply storing the lca for each pair of leaves in a table of size $O(n^2)$, and updating the table in time $O(n)$ each time a decision on some v_i is made (this does not hinder the total time complexity of the algorithm, though there are more clever ways to handle

dynamic tree *lca* queries [11]). Then the restrictions $S'_1|_q, \ldots, S'_h|_q$ can be computed in constant time. It is then straightforward to see that, by the above process, $\Pr[q \in Q(T')]$ can be computed in constant time. Each time a node v_i needs to be reinserted, this probability must be computed for the $O(n^3)$ quartets containing v_i. There are $n-1$ splits to be performed, and each split requires inserting $O(n)$ nodes. Thus the "binarization" process takes total time $O(n^5)$, and altogether the derandomization takes time $O(kn^4 + n^5)$.   ◀

The above leads to a (deterministic) 1/3- approximation. This can be used to show the following. The proof is similar to that of [9, Theorem 2] and is relegated to Appendix B.3.

▶ **Theorem 8.** *If WMQI can be approximated within a factor α, then WQC can be approximated within a factor $\beta = \alpha/(3\alpha - 2)$.*

Combined with Theorem 5 and letting $\alpha = 2$ in Theorem 8 we get the following.

▶ **Corollary 9.** *WQC can be approximated within a factor 1/2 in time $O(k^2n^2 + kn^4 + n^5)$.*

6 Fixed-parameter tractability of WQC

In this section we describe how, based on previous results on the minimum quartet incompatibility problem on complete sets, WQC can be solved in time $O(4^{d' + k'_2 + k'_3}n + n^4)$. Here k'_2 and k'_3 are the number of quadsets that have 2 and 3 dominant quartets, respectively, and d' is the number of strictly dominant quartets that we are allowed to reject. The algorithm makes direct use of the Gramm-Niedermeyer algorithm [14], henceforth called the GN algorithm.

The GN algorithm solves the following problem: given a *complete set* of quartets Q, find, if it exists, a complete and compatible set of quartets Q' such that at most d quartets of Q' are not in the input set Q (i.e. $|Q' \setminus Q| \le d$). This is accomplished by repeatedly applying the following theorem:

▶ **Theorem 10** ([14]). *Let Q be a complete set of quartets. Then Q is compatible if and only if for each set of five taxa $\{a, b, c, d, e\} \subseteq \mathcal{X}$, $ab|cd \in Q$ implies $ab|ce \in Q$ or $ae|cd \in Q$.*

The idea behind the GN algorithm is as follows: find a set of five taxa $\{a, b, c, d, e\}$ that does not satisfy the condition of Theorem 10, then correct the situation by branching into the four possible choices:
1. remove $ab|cd$ from Q and add $ac|bd$ to Q;
2. remove $ab|cd$ from Q and add $ad|bc$ to Q;
3. remove $\{ac|be, ae|bc\} \cap Q$ from Q and add $ab|ce$ to Q;
4. remove $\{ac|de, ad|ce\} \cap Q$ from Q and add $ae|cd$ to Q.

The quartets added to Q will not be questioned in the following branchings. With some optimization, this leads to a $O(4^d n + n^4)$ FPT algorithm.

In [14], the authors also note that this algorithm can be extended to sets of quartets Q that contain ambiguous quadsets, i.e. sets $\{a, b, c, d\}$ for which 2 or 3 of the possible quartets on $\{a, b, c, d\}$ are in Q. Suppose there are k_2 and k_3, respectively, quadsets that have 2 and 3 quartets in Q. The modified algorithm then, in a first phase, branches into the $2^{k_2}3^{k_3}$ ways of choosing one quartet per such quadset, thereby obtaining a complete set of quartets for each possibility. The GN algorithm is thus applied to the so-obtained complete sets. This yields a $O(2^{k_2} \cdot 3^{k_3} \cdot 4^d n + n^4)$ algorithm.

It is not hard to see that this gives an FPT algorithm for WQC, where the parameter k_2 (resp. k_3) is the number of quadsets such that 2 (resp. 3) possible quartets appear in the input trees, and d is the number of quartets $ab|cd$ that appear in every input tree, and that

we are allowed to discharge. Note however that, in the consensus setting, there is no reason to believe that k_2 and k_3 are low - in we fact we believe that $k_2 + k_3$ typically takes values in $\Theta(n^4)$. One reason is that even the slightest amount of noise on a quadset makes it included in the count of either k_2 or k_3 (e.g. if $k-1$ trees agree on $ab|cd$ and only one contains $ac|bd$).

The GN algorithm can, however, be used on a more interesting set of parameters. Define k_2' (resp. k_3') as the number of quadsets that have exactly 2 (resp. 3) dominant quartets, and let d' be the number of strictly dominant quartets that we are allowed to discharge. It is reasonable to believe that, if each tree of the input is close to the true tree T^*, most "true" quartets will appear as strictly dominant in the input, and there should not be too many ambiguous quadsets. There is a very simple algorithm achieving time $O(4^{d'+k_2'+k_3'}n + n^4)$. Construct a complete set of quartets Q as follows: for each quadset $\{a, b, c, d\}$, choose a dominant quartet on $\{a, b, c, d\}$ and add it to Q (if multiple choices are possible, choose arbitrarily). Then, run the GN algorithm on Q with the following modification: each time a quartet q is removed from Q and replaced by another quartet q', decrement either d', k_2' or k_3', depending on whether q belongs to a quadset with $1, 2$ or 3 dominant quartets. It follows that if there exists a complete and compatible set of quartets Q' such that at most d' strictly dominant quartets are rejected, then the modified algorithm will find it. It should be noted however that finding such a set Q' does not guarantee that the corresponding tree is an optimal solution. Indeed, since quartets are weighted, two solutions Q' and Q'' may both reject only d' strictly dominant quartets, yet one has higher weight than the other. However, the correctness of the algorithm follows from the fact that the GN algorithm finds the set of *every* solution discarding at most d' dominant quartets - and thus it suffices to pick the solution from this set that has optimal weight.

We finally mention that the FPT algorithms published in [10] are improved versions of the GN algorithm, can also return every solution and thus can be modified in the same manner. These yield FPT algorithms that can solve WQC in time $O(3.0446^{d'+k_2'+k_3'}n + n^4)$ and $O(2.0162^{d'+k_2'+k_3'}n^3 + n^5)$.

7 Conclusion

In this paper, we have shown that the WQC problem is NP-hard, answering a question of [19] and [2]. In the latter, the authors also propose a variant of the problem in which the output tree T is not required to be binary. In this case, one needs to assign a cost p to the unresolved quartets. Our reduction can be extended to show that hardness holds for high enough p, but the complexity of the general case remains open. We have also shown that WQC can be approximated within a factor $1/2$. One open question is whether the problem admits a PTAS as the related CMQC problem. The fixed-parameter tractability aspects of WQC also deserve further investigation. This would require identifying some structural properties that are present in the consensus setting and that can be used for designing practical exact algorithms. But as we have shown, this might not be an easy task, as many properties which seem reasonable for the consensus setting do not hold.

References

1 Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, and Jeffrey D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J. Comput.*, 10(3):405–421, 1981. `doi:10.1137/0210030`.

2 Mukul S. Bansal, Jianrong Dong, and David Fernández-Baca. Comparing and aggregating partially resolved trees. *Theor. Comput. Sci.*, 412(48):6634–6652, 2011. `doi:10.1016/j.tcs.2011.08.027`.

3 Jean-Pierre Barthélemy and Fred R. McMorris. The median procedure for n-trees. *J. Classif.*, 3(2):329–334, 1986. `doi:10.1007/BF01894194`.

4 Vincent Berry, David Bryant, Tao Jiang, Paul Kearney, Ming Li, Todd Wareham, and Haoyong Zhang. A practical algorithm for recovering the best supported edges of an evolutionary tree (extended abstract). In David B. Shmoys, editor, *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 287–296. ACM/SIAM, 2000. URL: `http://dl.acm.org/citation.cfm?id=338219.338265`.

5 Vincent Berry, Tao Jiang, Paul Kearney, Ming Li, and Todd Wareham. Quartet cleaning: Improved algorithms and simulations. In Jaroslav Nešetřil, editor, *Proceedings of the 7th Annual European Symposium on Algorithms (ESA 1999)*, volume 1643 of *LNCS*, pages 313–324. Springer Berlin Heidelberg, 1999. `doi:10.1007/3-540-48481-7_28`.

6 David Bryant. *Building trees, hunting for trees, and comparing trees.* PhD thesis, University of Canterbury, New Zealand, 1997. URL: `http://hdl.handle.net/10092/7914`.

7 David Bryant. A classification of consensus methods for phylogenetics. In Melvin F. Janowitz, François-Joseph Lapointe, Fred R. McMorris, Boris Mirkin, and Fred S. Roberts, editors, *Proceedings of DIMACS Working Group Meetings on Bioconsensus*, volume 61 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 163–184. Americal Mathematical Society, 2003. `doi:10.1090/dimacs/061/11`.

8 David Bryant and Mike Steel. Constructing optimal trees from quartets. *J. Algorithms*, 38(1):237–259, 2001. `doi:10.1006/jagm.2000.1133`.

9 Jaroslaw Byrka, Sylvain Guillemot, and Jesper Jansson. New results on optimizing rooted triplets consistency. *Discrete Appl. Math.*, 158(11):1136–1147, 2010. `doi:10.1016/j.dam.2010.03.004`.

10 Maw-Shang Chang, Chuang-Chieh Lin, and Peter Rossmanith. New fixed-parameter algorithms for the minimum quartet inconsistency problem. *Theory Comput. Syst.*, 47(2):342–367, 2010. `doi:10.1007/s00224-009-9165-y`.

11 Richard Cole and Ramesh Hariharan. Dynamic LCA queries on trees. *SIAM J. Comput.*, 34(4):894–923, 2005. `doi:10.1137/S0097539700370539`.

12 Joseph Felsenstein. *Inferring phylogenies.* Sinauer Associates Sunderland, 2004.

13 Zvi Galil and Nimrod Megiddo. Cyclic ordering is NP-complete. *Theor. Comput. Sci.*, 5(2):179–182, 1977. `doi:10.1016/0304-3975(77)90005-6`.

14 Jens Gramm and Rolf Niedermeier. Minimum quartet inconsistency is fixed parameter tractable. In Amihood Amir, editor, *Proceedings of the 12th Annual Symposium of Combinatorial Pattern Matching (CPM 2001)*, volume 2089 of *LNCS*, pages 241–256. Springer Berlin Heidelberg, 2001. `doi:10.1007/3-540-48194-X_23`.

15 Jesper Jansson. On the complexity of inferring rooted evolutionary trees. *Electron. Notes Discrete Math.*, 7:50–53, 2001. `doi:10.1016/S1571-0653(04)00222-7`.

16 Tao Jiang, Paul Kearney, and Ming Li. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM J. Comput.*, 30(6):1942–1961, 2001. `doi:10.1137/S0097539799361683`.

17 Timothy Margush and Fred R. McMorris. Consensus n-trees. *Bull. Math. Biol.*, 43(2):239–244, 1981. `doi:10.1007/BF02459446`.

18 Fred R. McMorris, David B. Meronk, and Dean A. Neumann. A view of some consensus methods for trees. In Joseph Felsenstein, editor, *Proceedings of the NATO Advanced Study Institute on Numerical Taxonomy*, volume 1 of *NATO Advanced Science Institutes Series, Series G: Ecological Sciences*, pages 122–126. Springer, 1983. `doi:10.1007/978-3-642-69024-2_18`.

19 Siavash Mirarab. *Novel scalable approaches for multiple sequence alignment and phylo-genomic reconstruction.* PhD thesis, University of Texas at Austin, 2015. URL: `http://hdl.handle.net/2152/31377`.

20 Siavash Mirarab, Rezwana Reaz, Md. Shamsuzzoha Bayzid, Théo Zimmermann, M. Shel Swenson, and Tandy Warnow. ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30(17):i541–i548, 2014. `doi:10.1093/bioinformatics/btu462`.

21 António Morgado and Joao Marques-Silva. A pseudo-boolean solution to the maximum quartet consistency problem, 2008. `arXiv:0805.0202`.

22 António Morgado and Joao Marques-Silva. Combinatorial optimization solutions for the maximum quartet consistency problem. *Fundam. Inform.*, 102(3-4):363–389, 2010. `doi:10.3233/FI-2010-311`.

23 Robert R. Sokal and F. James Rohlf. Taxonomic congruence in the leptopodomorpha re-examined. *Syst. Zool.*, 30(3):309–325, 1981. `doi:10.2307/2413252`.

24 Michael Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classif.*, 9(1):91–116, 1992. `doi:10.1007/BF02618470`.

25 Gang Wu, Jia-Huai You, and Guohui Lin. A lookahead branch-and-bound algorithm for the maximum quartet consistency problem. In Rita Casadio and Gene Myers, editors, *Proceedings of the 5th International Workshop on Algorithms in Bioinformatics (WABI 2005)*, volume 3692 of *LNCS*, pages 65–76. Springer, Springer, 2005. `doi:10.1007/11557067_6`.

26 Gang Wu, Jia-Huai You, and Guohui Lin. Quartet-based phylogeny reconstruction with answer set programming. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 4(1):139–152, 2007. `doi:10.1109/TCBB.2007.1008`.

A   Implications for the Weighted Triplet Consensus problem

For each set of three labels $\{a, b, c\}$, there are three non-isomorphic[3] rooted binary trees called *triplets.* They are denoted by $ab|c$, $ac|b$ and $bc|a$, depending on the leaf having the root as father (c, b and a respectively). We say that a tree T induces or displays the triplet $ab|c$ if $T|\{a, b, c\} = ab|c$. For a rooted tree R, denote by $tr(R)$ the set of triplets of R.

When the consensus is sought for rooted trees, the objective is to find a rooted tree M that induces a maximum number of triplets contained in the input trees. The Weighted Triplet Consensus (WTC) is defined as follows.

Weighted Triplet Consensus (WTC) problem
Input: a set of rooted trees $\mathcal{R} = \{R_1, \ldots, R_k\}$ such that $\mathcal{L}(R_1) = \ldots = \mathcal{L}(R_k) = \mathcal{X}$.
Output: a binary rooted tree M with $\mathcal{L}(M) = \mathcal{X}$ that maximizes $\sum_{R \in \mathcal{R}} |tr(M) \cap tr(R)|$.

As in the unrooted problem, other versions of WTC where the input trees may have missing species or where the weight of a triplet is not defined w.r.t. a set of trees, are known to be NP-hard [6]. The WTC problem is conjectured to be NP-hard in [2] (we note that a more general version where the output can be non-binary is also conjectured NP-hard).

We give the main idea behind the proof of the hardness of WTC. Let $\mathcal{T} = \bigcup_{C \in \mathcal{C}} \mathcal{T}(C)$ be the set of unrooted trees constructed in the reduction above. For a tree $T \in \mathcal{T}$, let e be the edge separating Z from the rest of the tree (i.e. by removing e from T, one connected component is exactly Z). Obtain a rooted tree R from T by rooting T at e, that is subdivide e, thereby creating a degree 2 vertex which is the root of R. The set of rooted trees $\mathcal{R}$ is obtained by applying this rooting to every $T \in \mathcal{T}$ (the Z subtree could be removed but we keep it here to make the correspondence easier to see).

---

[3] Isomorphism preserving labels and the root node.

Similarly as above, it can be shown that since every input tree is a rooted (W, Z)-caterpillar, then any solution must also have this form. This implies in turn that there exists a linear ordering of S satisfying $\mathcal{C}$ if and only if there is a solution M to WTC containing every triplet from the input on 2 or 3 members of $\mathcal{L}(W)$, every triplet containing at least one member of $\mathcal{L}(Z)$, plus at least $4m|W| + 3m|W| \left(\binom{n-2}{2} + 2(n-2) \right)$ triplets of the form $wa|b$ with $a, b \in S$. This is obtained by defining the notions of *in-triplets* and *out-triplets* analogously as in the previous section, but with respect to W only. That is, in a "$a < b$" tree, for $a, b, c, d \in S, w \in \mathcal{L}(W)$ and $\{c, d\} \neq \{a, b\}$, $wa|b$ would be an in-triplet, whereas $wc|d$ or $wd|c$ would be out-triplets. One can argue that for a cyclic triple $(a, b, c) \in \mathcal{C}$ and the set of trees $\mathcal{T}((a, b, c))$, an optimal consensus tree can contain $4|W|$ of the $6|W|$ possible in-triplets, plus at most half of the $6m|W| \left(\binom{n-2}{2} + 2(n-2) \right)$ possible out-triplets. The arguments are essentially the same as the ones given in the hardness proof of WQC, and so we omit the details.

▶ **Theorem 11.** *The* Weighted Triplet Consensus *problem is NP-hard.*

B Deferred proofs

B.1 Proof of Lemma 3

Despite the Lemma 3 statement being quite intuitive, it requires a surprising amount of care. We start by a simple proposition that will be needed.

▶ **Proposition 12.** *Let X, Y be two non-empty sets such that $Y \not\subseteq X$. Then $|X| \cdot |Y \setminus X| \geq |Y| - 1$.*

Proof. Suppose first that $X \cap Y = \emptyset$. Then clearly $|X||Y \setminus X| = |X||Y| \geq |Y| - 1$. Suppose otherwise that $X \cap Y \neq \emptyset$, and denote $X' = X \cap Y$. Then $|Y \setminus X| = |Y| - |X'|$ and since $Y \not\subseteq X$, we must have $|Y| \geq |X'| + 1$. We also have $|X||Y \setminus X| = |X|(|Y| - |X'|) \geq |X'|(|Y| - |X'|)$; we claim the latter term to be at least $|Y| - 1$. Let us assume for contradiction that $|X'|(|Y| - |X'|) < |Y| - 1$. If $|X'| = 1$, this is clearly impossible, so assume $|X'| > 1$. Then we get $|X'||Y| - |Y| < |X'|^2 - 1$ leading to $|Y| < \frac{|X'|^2 - 1}{|X'| - 1} = |X'| + 1$, contradicting $|Y| \geq |X'| + 1$. ◀

Before proceeding, we must introduce the notion of a rooted subtree of a binary unrooted tree T. Note that by removing an edge $e = \{u, v\}$ of T, we obtain two disjoint rooted subtrees T_1 and T_2, respectively rooted at u and v. Call T' a *rooted subtree* of T if T' is a rooted tree that can be obtained by removing an edge of T. For $X \subset \mathcal{L}(T)$, a *rooted subtree for X* is a rooted subtree T' of T such that $X \subseteq \mathcal{L}(T')$. We denote by $T[X]$ the rooted subtree for X that contains a minimum number of leaves (if there are multiple choices, choose $T[X]$ arbitrarily among the possible choices). Note that $T[X]$ may contain leaves other than X.

We now prove that any optimal solution to $\mathcal{T}$ as constructed in our reduction must be a (W, Z)-augmented caterpillar. Suppose that M is an optimal solution for $\mathcal{T}$, and that M is not a (W, Z)-augmented caterpillar. Denote $M_W = M[\mathcal{L}(W)]$ and $M_Z = M[\mathcal{L}(Z)]$. If M is a (W', Z')-augmented caterpillar $(W'|T_1|\ldots|T_k|Z')$ for some trees W', Z' with $\mathcal{L}(W') = \mathcal{L}(W)$ and $\mathcal{L}(Z') = \mathcal{L}(Z)$, it is not hard to see that $M' = (W|T_1|\ldots|T_k|Z)$ is a better solution than M, a contradiction. Thus, M is not such a caterpillar, and this implies that either $\mathcal{L}(M_W) \neq \mathcal{L}(W)$ or $\mathcal{L}(M_Z) \neq \mathcal{L}(Z)$ (or both). That is, the rooted subtrees containing $\mathcal{L}(W)$ and/or $\mathcal{L}(Z)$ have "outsider" leaves. Suppose first that $\mathcal{L}(M_W) \neq \mathcal{L}(W)$ holds. Then there exists a node x with children x_l and x_r in M_W such that all leaves X_l below x_l are in $\mathcal{L}(W)$ with $\mathcal{L}(W) \not\subseteq X_l$ (otherwise $M_W = M[\mathcal{L}(W)]$ would be chosen incorrectly), and no leaf X_r

below x_r belongs to $\mathcal{L}(W)$ (this can be seen by observing that the minimal node x of M_W having leaves both in W and not in W has this property).

We claim that $\mathcal{L}(Z) \not\subseteq X_r$. Suppose otherwise that $\mathcal{L}(Z) \subseteq X_r$. Then $|X_r| \geq |Z|$ and so $|M_W| \geq |W| + |Z|$. However in M, by removing the xx_r edge we obtain two rooted trees, one of which is a rooted subtree for $\mathcal{L}(W)$. Moreover, this subtree has at most $|W| + |S| < |W| + |Z|$ leaves, which contradicts the minimality of $M_W = M[\mathcal{L}(W)]$. We deduce that $\mathcal{L}(Z)$ is not a subset of X_r.

Now, observe that M contains the quartet $w_1y|w_2z$ for each $w_1 \in X_l, y \in X_r, w_2 \in \mathcal{L}(W) \setminus X_l, z \in \mathcal{L}(Z) \setminus X_r$. There are at least $|X_l||X_r|(|\mathcal{L}(W) \setminus X_l|)(|\mathcal{L}(Z) \setminus X_r|) \geq (|W|-1)(|Z|-1)$ such quartets (the inequality is obtained by applying Proposition 12 to $|X_l| \cdot |\mathcal{L}(W) \setminus X_l|$ and $|X_r| \cdot |\mathcal{L}(Z) \setminus Z|$). Moreover, each input tree of $\mathcal{T}$ contains the quartet $w_1w_2|yz$ instead, and hence in total in $\mathcal{T}$ there are at least $6m(|W|-1)(|Z|-1)$ quartets of the form $w_1w_2|yz$ that M does not contain. In the same manner, if the case $\mathcal{L}(M_Z) \neq \mathcal{L}(Z)$ holds, then there are at least $6m(|W|-1)(|Z|-1)$ quartets of the form $z_1z_2|yw$ that M does not contain, where here $z_1, z_2 \in \mathcal{L}(Z), y \notin \mathcal{L}(Z), w \in \mathcal{L}(W)$.

Now, let $\rho(M)$ be the number of quartets that M contains from $\biguplus_{T \in \mathcal{T}} Q(T)$ that have the form $wx|yz$, where $w \in \mathcal{L}(W), z \in \mathcal{L}(Z), x, y \in S$. Formally,

$$\rho(M) = \sum_{\substack{wx|yz \in Q(M) \\ x,y \in S \\ w \in \mathcal{L}(W) \\ z \in \mathcal{L}(Z)}} f(wx|yz)$$
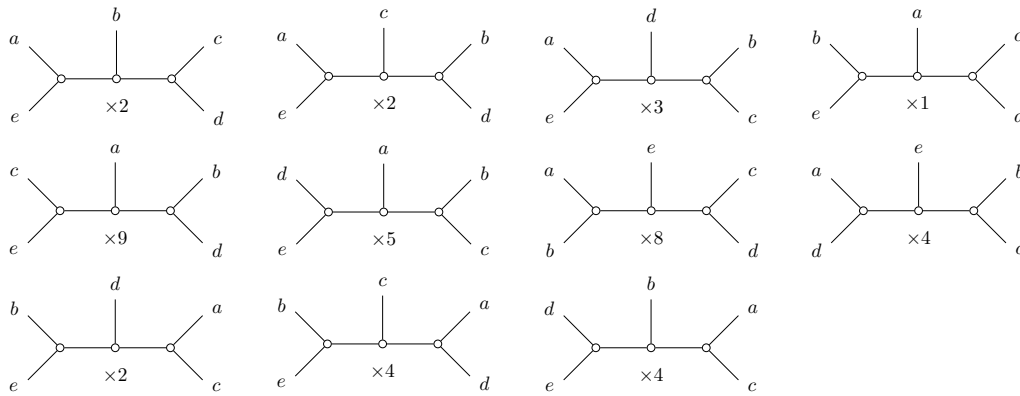
where $f(wx|yz)$ denotes the number of trees of $\mathcal{T}$ that contain the $wx|yz$ quartet. For a given $u \in \mathcal{L}(W) \cup \mathcal{L}(Z)$, let $\rho(M, u)$ denote the number of quartets counted in $\rho(M)$ that contain u. Formally, if $w \in \mathcal{L}(W)$, we have

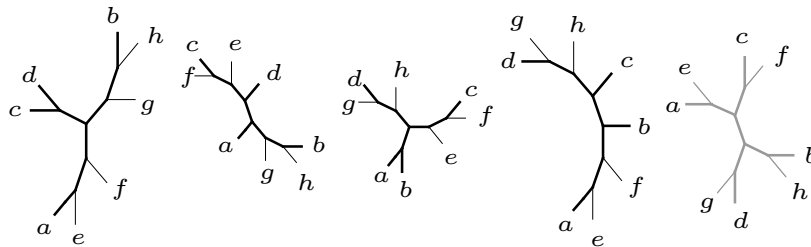$$\rho(M, w) = \sum_{\substack{wx|yz \in Q(M) \\ x,y \in S \\ z \in \mathcal{L}(Z)}} f(wx|yz) \,.$$

The definition of $\rho(M, z)$ is the same for $z \in \mathcal{L}(Z)$, except that z gets fixed instead of w in the summation. Notice that $\rho(M) = \sum_{w \in \mathcal{L}(W)} \rho(M, w) = \sum_{z \in \mathcal{L}(Z)} \rho(M, z)$. Let $w^* = \arg\max_{w \in \mathcal{L}(W)} \{\rho(M, w)\}$. We obtain an alternative solution M' from M in the following manner: remove all leaves of $\mathcal{L}(W) \setminus \{w^*\}$ from M, delete the degree 2 nodes, and replace w^* by the W tree. Note that if $w^*x|yz$ is a quartet of M, then $wx|yz$ is a quartet of M' for all $w \in \mathcal{L}(W)$, and so $\rho(M', w) \geq \rho(M, w)$ for all such w by the choice of w^*. Consequently, $\rho(M') \geq \rho(M)$. We repeat the same operation on M' for the Z tree and obtain our final tree M^*. That is, we find $z^* = \arg\max_{z \in \mathcal{L}(Z)} \{\rho(M', z)\}$, and replace z^* by the Z tree. As above, we obtain $\rho(M^*) \geq \rho(M')$. Since M^* has W and Z as rooted subtrees, it follows that M^* is a (W, Z)-augmented caterpillar.

We argue that M^* contains more quartets from the input trees than M. First observe that the quartets on which M and M^* differ must contain a member of $\mathcal{L}(W) \cup \mathcal{L}(Z)$, since only these leaves switched position. The tree M^* contains every quartet of $\biguplus_{T \in \mathcal{T}} Q(T)$ that have at least two members of $\mathcal{L}(W)$, or two members of $\mathcal{L}(Z)$. This includes the aforementioned (at least) $6m(|W|-1)(|Z|-1)$ quartets of the form $w_1w_2|yz$ or $z_1z_2|yw$ that M does not contain. As for the quartets that contain one member of $\mathcal{L}(W)$ and one member of $\mathcal{L}(Z)$, M^* contains at least as many such quartets as M since in $\biguplus_{T \in \mathcal{T}} Q(T)$, these quartets are all of the form $wx|yz$, and we have $\rho(M^*) \geq \rho(M)$. Finally, each tree of $\mathcal{T}$ has at most $(|W| + |Z|)n^3$ quartets that have exactly one member of $\mathcal{L}(W) \cup \mathcal{L}(Z)$. Thus

Figure 1 An instance of WQC such that the optimal solution (the third tree on the first row) contains no strictly dominant quartet. The numbers correspond to the number of times that each tree appears in the input.

Figure 2 The first four trees form an instance of WQC in which every tree contains $ab|cd$. The rightmost tree is the unique optimal solution to the WQC instance (every possible solution was verified computationally).

at most $6m(|W| + |Z|)n^3$ quartets of this type are contained by M and not contained by M^*, but since this is smaller than $6m(|W| - 1)(|Z| - 1)$ for our choice of $|W|$ and $|Z|$, M^* contains more quartets from the input than M.

B.2 Proofs of Section 4

Conjecture 1 is disproved by Theorem 13, and Conjecture 3 by Theorem 14, which implies that Conjectures 2 and 4 are also false; finally Conjecture 5 is disproved by Theorem 15.

▶ **Theorem 13.** *There exists an instance of WQC such that every optimal solution contains none of the strictly dominant quartets.*

Figure 1 shows an instance of WQC demonstrating Theorem 13. In this instance, for every quadset S, there is a strictly dominant quartet appearing 17 times, whereas the second-most and third-most quartets appear in 16 and 11 trees, respectively. For example, $f(ac|bd) = 17$, $f(ad|bc) = 16$ and $f(ab|cd) = 11$. One can check that the best tree is the third one on the top row (the $ae|bc$ with d grafted on the middle branch). Call this tree T^*. For every quadset S, T^* contains the second-most frequent quartet on S. The reason why T^* is optimal is that, in the particular instance of Figure 1, any other tree T that contains a strictly dominant quartet for some quadset S must also contain a least frequent quartet on

some other quadset S'. Hence, as there are 5 quadsets, T contains at most $4 \cdot 17 + 11 = 79$ quartets from the input, whereas T^* contains $5 \cdot 16 = 80$. Note that this example consists of trees on only 5 leaves. We do not know if such instances exist for any $n > 5$ leaves.

▶ **Theorem 14.** *There exists an instance of WQC such that there is a quartet q that appears in every input tree, but q is not a quartet of any optimal solution.*

Figure 2 shows an instance of WQC proving Theorem 14. Each input tree contains the $ab|cd$ quartet, whereas the optimal solution, which is unique, does not. The rightmost tree contains 180 quartets from the input multiset $\mathcal{Q}$, whereas any other tree has at most 176.

Finally, we note that the main interest behind Conjecture 5 is the following: if it holds, in cases where the set F of strictly least-frequent quartets is complete we could tell in polynomial time – using results of [8] – whether there is a tree T^* that contains no quartet from F. Conjecture 5 could then lead to interesting approximations or FPT algorithms. However, least-frequent quartets cannot be excluded automatically.

▶ **Theorem 15.** *There exists an instance of WQC such that every optimal solution contains a strictly least-frequent quartet, even if there exists a tree T^* with no such quartet.*

The instance corresponding to Theorem 15 is obtained from the instance shown in Figure 1, by removing all occurrences of the third tree on the top row (i.e. this tree now appears 0 times instead of 3 times). The second-most frequent quartets now appear 13 times each, and so the tree T^* that contains all these quartets has a total weight of $5 \cdot 13 = 65$. However, there are trees with a total weight of 75, which are optimal (for instance, the tree of cardinality 9 in the figure). Each such tree contains a strictly dominant quartet, and as mentioned before, also a strictly least-frequent quartet.

B.3   Proof of Theorem 8

Let $N := k\binom{n}{4}$, i.e. the total number of quartets in $\mathcal{Q}$, let p be the maximum number of quartets that can be preserved from $\mathcal{Q}$ for compatibility, and let d be the minimum number of quartets to discard from $\mathcal{Q}$ in order to attain compatibility (here p and d refer to multiset cardinalities). Note that $d = N - p$. We show that taking the best tree between the one obtained from the factor α algorithm for WMQI and the one obtained from the "random-tree-algorithm" achieves a factor β for WQC. Suppose first that $p \leq N/(3\beta)$. By Lemma 7, the "random-tree-algorithm" yields a tree containing at least $|\mathcal{Q}|/3 = N/3$ quartets from $\mathcal{Q}$, and since $N/3 = \beta N/(3\beta) \geq \beta p$, it yields a solution to WQC within a factor β from optimal. Thus we may assume that $p > N/(3\beta) = N(3\alpha - 2)/(3\alpha)$. Since we have an α approximation for WMQI, we may obtain a solution discarding at most $\alpha d = \alpha(N - p)$ quartets. This solution preserves at least $N - (\alpha(N - p)) = \alpha p + (1 - \alpha)N$ quartets from $\mathcal{Q}$. We claim that this attains a factor β approximation. Suppose instead that $\alpha p + (1 - \alpha)N < \beta p$. Then $p < (\alpha - 1)N/(\alpha - \beta)$ which, with a little work, yields $p < N(3\alpha - 2)/(3\alpha)$, contradicting our assumption on p. Thus, the WMQI approximation preserves at least βp quartets.