A simple 4-approximation algorithm for maximum agreement forests on multiple unrooted binary trees

Dempsey, Jordan; van Iersel, Leo; Jones, Mark; Zeh, Norbert

DOI
10.1016/j.ipl.2025.106572

Publication date
2025

Document Version
Final published version

Published in
Information Processing Letters

Inf. Process. Lett. 190 (2025) 106572

Contents lists available at ScienceDirect

Information Processing Letters

journal homepage: www.elsevier.com/locate/ipl

A simple 4-approximation algorithm for maximum agreement forests on multiple unrooted binary trees ☆

Jordan Dempsey [a,*], Leo van Iersel [b], Mark Jones [b], Norbert Zeh [a]

[a] Faculty of Computer Science, Dalhousie University, 6050 University Avenue, Halifax, B3H 1W5, Nova Scotia, Canada
[b] Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, Delft, 2628 CD, the Netherlands

ARTICLE INFO

Keywords:
Phylogenetic trees
TBR distance
Agreement forests
Approximation algorithms
Linear programming
LP rounding

ABSTRACT

Maximum agreement forests have been used as a measure of dissimilarity of two or more phylogenetic trees on a given set of taxa. An agreement forest is a set of trees that can be obtained from each of the input trees by deleting edges and suppressing degree-2 vertices. A maximum agreement forest is such a forest with the minimum number of components. We present a simple 4-approximation algorithm for computing a maximum agreement forest of multiple unrooted binary trees. This algorithm applies LP rounding to an extension of a recent ILP formulation of the maximum agreement forest problem on two trees by Van Wersch et al. [13]. We achieve the same approximation ratio as the algorithm by Chen et al. [3], but our algorithm is extremely simple. We also prove that no algorithm based on the ILP formulation by Van Wersch et al. can achieve an approximation ratio of $4-\varepsilon$, for any $\varepsilon > 0$, even on two trees. To this end, we prove that the integrality gap of the ILP approaches 4 as the size of the two input trees grows.

1. Introduction

Phylogenetic trees (and networks) model the evolution of a set of taxa. Different methods for constructing such trees from, say, DNA data may produce different results. Even the same method may produce different results when trees on the same set of taxa are constructed from different genes shared by these taxa. As a result, it has become important to measure the dissimilarity between phylogenetic trees, both to quantify confidence in the trees constructed using different methods and to discover non-tree-like events in the evolution of a set of taxa that explain the differences between trees constructed from different genes.

One distance measure used to quantify the dissimilarity between two unrooted phylogenetic trees is the *tree bisection and reconnection* (TBR) distance [1]. A TBR operation on a tree T removes an edge $\{u, v\}$ of T, thereby splitting T into two subtrees T_u and T_v; supresses u an v, as they now have degree 2; subdivides an edge in T_u and an edge in T_v; and then reconnects T_u and T_v by adding an edge between the two vertices introduced by subdividing these two edges. The TBR distance between two trees is the number of such TBR operations necessary to turn one of the two trees into the other. This distance is known to be one less than the size of a *maximum agreement forest (MAF)* of the two trees [1].

An *agreement forest* (AF) of a set of trees $\mathcal{T}$ is a forest that can be obtained from each tree in $\mathcal{T}$ by deleting edges and suppressing degree-2 vertices. A *maximum agreement forest* (MAF) is an agreement forest with the minimum number of components (which corresponds to preserving the maximum number of edges in each tree in $\mathcal{T}$). While the TBR distance is difficult to extend to more than two trees, the definition of a MAF does generalize naturally to more than two trees and is meaningful as a measure of (dis)similarity of the given set of trees, as it captures the parts of the evolutionary history of a set of taxa on which all input trees agree. While agreement forests have also been studied for rooted trees (e.g., [3,12,14]), we focus on unrooted trees in this paper.

Computing a MAF is NP-hard even for two trees [1,8]. This motivates the study of parameterized and approximation algorithms for computing MAFs. The best known kernel for this problem, due to Kelk et al. [9], has size $9k-8$, where k is the size of the MAF. Hallett and McCartin provided a branching algorithm for the same problem with running time $O(4^k \cdot k^5 + n^{O(1)})$ [7]. Chen et al. further improved this bound to $O(3^k \cdot n)$ [2]. Van Wersch et al. provided a new ILP formulation of the MAF problem, as well as improved kernelization results that were incorporated into the Tubro software for computing TBR distance [13].

☆ This paper received funding from the Netherlands Organisation for Scientific Research (NWO) under projects OCENW.M.21.306 and OCENW.KLEIN.125, and from the Natural Sciences and Engineering Research Council of Canada under grant RGPIN/05435-2018.

* Corresponding author.
E-mail address: J.Dempsey@dal.ca (J. Dempsey).

https://doi.org/10.1016/j.ipl.2025.106572
Received 12 September 2024; Received in revised form 17 February 2025; Accepted 27 February 2025
Available online 28 February 2025

Whidden and Zeh presented a linear-time 3-approximation algorithm for the unrooted MAF problem on two trees [15], which remains the best known algorithm for this problem. Chen et al. [3] presented a 4-approximation algorithm for multiple binary trees. Their algorithm is purely combinatorial but is rather complicated — significantly more complicated than the 3-approximation algorithm for two trees.

In this paper, we show that the ILP formulation of the MAF problem by Van Wersch et al. [13] can be combined with an extremely simple LP rounding approach to match the approximation ratio achieved by Chen et al. on multiple binary trees. We also show that no algorithm based on this ILP formulation can achieve an approximation ratio of $4-\varepsilon$ for any constant $\varepsilon > 0$, even when restricted to two trees. We do this by proving that the integrality gap of this ILP formulation approaches 4 as the size of the two input trees grows.

The remainder of this paper is organized as follows: Section 2 provides formal definitions of the concepts used in this paper. Section 3 provides an ILP formulation of the problem of computing a MAF of a set of trees. Section 4 provides a 4-approximation algorithm for computing a MAF of a set of trees based on this ILP formulation. Section 5 proves that the integrality gap of this ILP is $4 - o(1)$, thus precluding a better approximation ratio than 4 for any algorithm based on this ILP. Section 6 offers concluding remarks and explains the significance of our results for related problems.

2. Preliminaries

A *(binary) phylogenetic tree* T is a tree whose internal vertices have degree 3 and are unlabelled, and whose leaves are labelled bijectively with the elements of some set X. We call the elements of X *taxa* and do not distinguish between a leaf and its label. All trees in this paper are unrooted, that is, they are connected undirected graphs without cycles.

Two phylogenetic trees T_1 an T_2 over the same leaf set X are *isomorphic* (written $T_1 \cong T_2$) if there exists a graph isomorphism $\phi : T_1 \to T_2$ such that $\phi(x) = x$, for all $x \in X$ (i.e., ϕ respects the leaf labels).

For any subset $Y \subseteq X$ we denote by $T[Y]$ the minimal subtree of T that connects all leaves in Y. We use $T|_Y$ to refer to the tree obtained by suppressing all degree-2 vertices in $T[Y]$. *Suppressing* a degree-2 vertex v with neighbours u and w is the operation of removing v and its incident edges and reconnecting u and w with an edge $\{u, w\}$.

For a set $\mathcal{T} = \{T_1, \dots, T_t\}$ of phylogenetic trees over the same leaf set X, an *agreement forest* (AF) of $\mathcal{T}$ is a partition $\mathcal{F} = \{Y_1, \dots, Y_k\}$ of X such that[1]

1. For all $1 \le i \le j \le t$ and $1 \le h \le k$, $T_i|_{Y_h} \cong T_j|_{Y_h}$ and
2. For all $1 \le i \le t$ and $1 \le h < h' \le k$, $T_i[Y_h]$ and $T_i[Y_{h'}]$ are disjoint subtrees of T_i.

This captures the intuitive definition of an AF given in the introduction: Condition (2) expresses that the trees $T_i[Y_1], \dots, T_i[Y_t]$ are separated by edges in T_i, so the set of these trees can be obtained from T_i by cutting some set of edges in T_i. Condition (1) expresses that we must obtain the same collection of trees from every tree in $\mathcal{T}$ after suppressing degree-2 vertices. We call $Y_1, \dots, Y_k$ the *components* of $\mathcal{F}$. We say that two distinct components $Y_h, Y_{h'} \in \mathcal{F}$ *overlap in* T_i if they violate condition (2) for T_i (in this case, $\mathcal{F}$ is not an AF of $\mathcal{T}$). We say that $\mathcal{F}$ is a *maximum agreement forest* (MAF) of $\mathcal{T}$ if there is no AF $\mathcal{F}'$ of $\mathcal{T}$ of size $|\mathcal{F}'| < |\mathcal{F}|$.

A *quartet* is a subset of X of size 4. For a quartet $Q = \{a, b, c, d\}$, let $ab|cd$ be the tree with leaf set Q in which a and b share a common neighbour u, c and d share a common neighbour v, and u and v are connected by an edge. If $T_1|_Q \cong ab|cd$, then we define $\mathcal{L}(Q)$ to be the set of edges in $T_1[\{a, b\}] \cup T_1[\{c, d\}]$. A quartet $Q \subseteq X$ is an *incompatible* quartet of two trees $T_i, T_j \in \mathcal{T}$ if $T_i|_Q \ncong T_j|_Q$.

Lemma 2.1. *Two phylogenetic trees T_1 and T_2 on the same leaf set X are isomorphic if and only if they have no incompatible quartets.*

Lemma 2.1 follows from the work of Colonius and Schulze [4,5].

3. An ILP formulation of the MAF problem

Let $\mathcal{T} = \{T_1, \dots, T_t\}$ be a set of t phylogenetic trees over the same label set X. For $2 \le i \le t$, let $\mathcal{Q}_i$ be the set of incompatible quartets of T_1 and T_i. Let $\mathcal{Q} = \bigcup_{i=2}^{t} \mathcal{Q}_i$. We prove that an optimal solution to the following ILP defines a MAF of $\mathcal{T}$:

$$\begin{aligned} &\text{Minimize} \sum_{e \in E(T_1)} x_e \\ \text{s.t.} &\sum_{e \in \mathcal{L}(Q)} x_e \ge 1 \qquad \forall Q \in \mathcal{Q} \\ &x_e \in \{0, 1\} \quad \forall e \in E(T_1). \end{aligned} \tag{1}$$

Any solution $\hat{x}$ of (1) defines a set $E_{\hat{x}} = \{e \in E(T_1) \mid \hat{x}_e = 1\}$. The mapping $\hat{x} \mapsto E_{\hat{x}}$ is easily seen to be a bijection between the set of solutions of (1) and the set of subsets of $E(T_1)$. Thus, we mostly do not distinguish between solutions of (1) and subsets of $E(T_1)$.

Any subset of edges $E \subseteq E(T_1)$ defines a partition $\mathcal{F}_E = \{Y_1, \dots, Y_k\}$ of X where two leaves $a, b \in X$ belong to the same component Y_h if and only if $T_1[\{a, b\}] \cap E = \emptyset$.

Theorem 3.1. *A subset $E \subseteq E(T_1)$ is a feasible solution of (1) if and only if $\mathcal{F}_E$ is an agreement forest of $\mathcal{T}$.*

Proof. The proof follows the proof for two trees [13].

First assume $\mathcal{F}_E$ is not an AF of $\mathcal{T}$. Then there exist either a tree $T_i \in \mathcal{T}$ and two components $Y_h, Y_{h'} \in \mathcal{F}_E$ that overlap in T_i, or two trees $T_i, T_j \in \mathcal{T}$ and a component $Y_h \in \mathcal{F}_E$ such that $T_i|_{Y_h} \ncong T_j|_{Y_h}$.

In the latter case, we can assume w.l.o.g. that $T_1|_{Y_h} \ncong T_i|_{Y_h}$ because $T_i|_{Y_h} \ncong T_j|_{Y_h}$ implies that we cannot have both $T_1|_{Y_h} \cong T_i|_{Y_h}$ and $T_1|_{Y_h} \cong T_j|_{Y_h}$. By Lemma 2.1, this implies that there exists a quartet $Q \subseteq Y_h$ with $T_1|_Q \ncong T_i|_Q$. Since $Q \subseteq Y_h$, we have $\mathcal{L}(Q) \cap E = \emptyset$, so E is not a feasible solution of (1).

If two components $Y_h, Y_{h'} \in \mathcal{F}_E$ overlap in T_i, then there exist two leaves $a, b \in Y_h$ and two leaves $c, d \in Y_{h'}$ such that the two paths $T_i[\{a, b\}]$ and $T_i[\{c, d\}]$ share an edge. Thus, for $Q = \{a, b, c, d\}$, $T_i|_Q \cong ac|bd$ or $T_i|_Q \cong ad|bc$. On the other hand, $T_1[\{a, b\}] \cap E = \emptyset$ and $T_1[\{c, d\}] \cap E = \emptyset$ because $a, b \in Y_h$ and $c, d \in Y_{h'}$, and $T_1[\{x, y\}] \cap E \neq \emptyset$, for all $x \in \{a, b\}$ and $y \in \{c, d\}$ because $x \in Y_h$, $y \in Y_{h'}$, but $Y_h \neq Y_{h'}$. This implies that $T_1|_Q \cong ab|cd \ncong T_i|_Q$ and that $\mathcal{L}(Q) \cap E = \emptyset$, so once again, E is not a feasible solution of (1).

Now assume E is not a feasible solution of (1). Then there exists a quartet $Q = \{a, b, c, d\} \in \mathcal{Q}$ such that $T_1|_Q \cong ab|cd$ and $E \cap \mathcal{L}(Q) = \emptyset$. Assume $Q \in \mathcal{Q}_i$. Since $E \cap \mathcal{L}(Q) = \emptyset$, a and b belong to the same component Y_h of $\mathcal{F}_E$, and c and d belong to the same component $Y_{h'}$ of $\mathcal{F}_E$. Since $T_1|_Q \cong ab|cd$ and Q is an incompatible quartet of T_1 and T_i, the paths $T_i[\{a, b\}]$ and $T_i[\{c, d\}]$ share an edge. Thus, if $Y_h \neq Y_{h'}$, these two components overlap in T_i, and $\mathcal{F}_E$ is not an AF of $\mathcal{T}$. If $Y_h = Y_{h'}$, then $T_i|_Q \cong ac|bd$ or $T_i|_Q \cong ad|bc$. In either case, $T_1|_Q \ncong T_i|_Q$. Thus, by Lemma 2.1, $T_1|_{Y_h} \ncong T_i|_{Y_h}$ and, once again, $\mathcal{F}_E$ is not an AF of $\mathcal{T}$. □

It is easily verified that every AF $\mathcal{F}$ of $\mathcal{T}$ satisfies $\mathcal{F} = \mathcal{F}_E$, for some subset of edges $E \subseteq E(T_1)$, that any such set has size $|E| \ge |\mathcal{F}| - 1$, and that there exists such a set of size $|E| = |\mathcal{F}| - 1$. Thus, since $|E_{\hat{x}}| = \sum_{e \in E(T_1)} \hat{x}_e$, Theorem 3.1 immediately implies the following corollary.

Corollary 3.2. *A subset $E \subseteq E(T_1)$ is an optimal solution of (1) if and only if $\mathcal{F}_E$ is a maximum agreement forest of $\mathcal{T}$ and $|E| = |\mathcal{F}_E| - 1$.*

[1] This definition follows Linz and Semple [11]. Replacing $\mathcal{F} = \{Y_1, \dots, Y_2\}$ with $\mathcal{F} = \{T_1|_{Y_1}, \dots, T_1|_{Y_2}\}$ produces the equivalent definition used by Allen and Steel [1], where $\mathcal{F}$ is an actual forest.

4. A 4-approximation based on LP rounding

Now consider the LP relaxation of (1), where the constraint $x_e \in \{0,1\}$ is replaced with the constraint $x_e \geq 0$. Let $\tilde{x}$ be an optimal fractional solution of this LP relaxation. By Theorem 3.1, any integral feasible solution $\hat{x}$ of (1) corresponds to an AF $\mathcal{F}_{E_{\hat{x}}}$ of $\mathcal{T}$ of size $|\mathcal{F}_{E_{\hat{x}}}| \leq |E_{\hat{x}}|+1 = \sum_{e\in E(T_1)} \hat{x}_e + 1$. By Corollary 3.2, any optimal integral solution x^* of (1) corresponds to a MAF $\mathcal{F}_{E_{x^*}}$ of size $|\mathcal{F}_{E_{x^*}}| = |E_{x^*}|+1 = \sum_{e\in E(T_1)} x^*_e + 1 \geq \sum_{e\in E(T_1)} \tilde{x}_e + 1$. Thus, if $\sum_{e\in E(T_1)} \hat{x}_e \leq 4\sum_{e\in E(T_1)} \tilde{x}_e$, then $|\mathcal{F}_{E_{\hat{x}}}| \leq 4|\mathcal{F}_{E_{x^*}}|$, that is, $\mathcal{F}_{E_{\hat{x}}}$ is a 4-approximation of a MAF of $\mathcal{T}$. Such a solution can be obtained as follows. It is more intuitive to describe the construction in terms of the edge set $E = E_{\hat{x}}$.

We compute an optimal fractional solution $\tilde{x}$ of (1), choose an arbitrary leaf r of T_1 as its root, and initially set $E = \emptyset$. For every edge e, we name its endpoints u_e and v_e such that u_e is on the path from r to v_e (i.e., u_e is the parent of v_e). Let $D(e)$ be the set of descendant edges of e that belong to the same connected component of $T_1 - E$ as v_e. Formally, $f \in D(e)$ if v_e belongs to the path from r to v_f and the path from v_e to v_f contains no edge in E. Note that e itself meets these conditions, so $e \in D(e)$. Finally, let $w(e) = \sum_{f\in D(e)} \tilde{x}_f$. Now we choose an edge e such that

$$w(e) \geq 1/4 \text{ but } w(f) < 1/4 \ \forall f \in D(e) \setminus \{e\} \qquad (*)$$

and add this edge to E (if such an edge exists). Note that this changes the values of $D(f)$ and $w(f)$ for every edge f on the path from r to v_e. We continue adding edges that satisfy (*) to E until every edge $e \in E(T_1) \setminus E$ satisfies $w(e) < 1/4$. At the end of the algorithm, we define D_r to be the set of edges f such that the path from r to v_f contains no edge in E. This set may be empty and is used only in the analysis of the algorithm.

Next we prove that $|E| \leq 4\sum_{e\in E(T_1)} \tilde{x}_e$ and that E is a feasible solution of (1). As argued above, this implies that $\mathcal{F}_E$ is a 4-approximation of a MAF of $\mathcal{T}$.

Lemma 4.1. $|E| \leq 4 \sum_{e\in E(T_1)} \tilde{x}_e$.

Proof. For every edge $e \in E$, let $D_1(e)$ and $D_2(e)$ be the values of $D(e)$ at the time when e is added to E and when the algorithm terminates, respectively. Let $w_1(e) = \sum_{f\in D_1(e)} \tilde{x}_f$. Note that $D_2(e_1) \cap D_2(e_2) = \emptyset$, for any two distinct edges $e_1, e_2 \in E$.

For every edge $e \in E$ and every edge $f \in D_2(e) \setminus \{e\}$, the path from v_e to v_f contains no edge in E when the algorithm terminates. Thus, this is also true at the time when the algorithm adds e to E. This shows that $f \in D_1(e)$, that is, $D_2(e) \subseteq D_1(e)$.

If $D_1(e) \nsubseteq D_2(e)$, then there exists an edge $f \in D_1(e) \cap E$ that is added to E after e. We have $w(f) \geq 1/4$ at the time we add f to E and, since $f \in D_1(e)$, $w(f) < 1/4$ at the time we add e to E. Adding edges to E cannot increase $w(f)$ for any edge $f \in E(T_1)$. Thus, $w(f)$ never increases. This is a contradiction, and we must have $D_1(e) \subseteq D_2(e)$ and, therefore, $D_1(e) = D_2(e)$.

Since $D_2(e_1) \cap D_2(e_2) = \emptyset$, for any two distinct edges $e_1, e_2 \in E$, we also have $D_1(e_1) \cap D_1(e_2) = \emptyset$, for any two such edges e_1 and e_2. This implies that $\sum_{e\in E} w_1(e) = \sum_{e\in E}\sum_{f\in D_1(e)} \tilde{x}_f \leq \sum_{e\in E(T_1)} \tilde{x}_e$.

On the other hand, we have $w_1(e) \geq 1/4$, for every edge $e \in E$. Thus, $\sum_{e\in E} w_1(e) \geq |E|/4$, that is, $|E| \leq 4\sum_{e\in E(T_1)} \tilde{x}_e$. □

Lemma 4.2. *E is a feasible solution of* (1).

Proof. Similar to the proof of Lemma 4.1, consider the values of $D(e)$ and $w(e)$ at the end of the algorithm, for every edge $e \in E(T_1)$. Then $w(e) < 1/4$, for every edge $e \notin E$.

For any two leaves $a, b \in X$, let u be their lowest common ancestor in T_1, that is, the vertex closest to r that belongs to $T_1[\{a,b\}]$. Then $T_1[\{a,b\}] = T_1[\{a,u\}] \cup T_1[\{u,b\}]$ and $\sum_{e\in T_1[\{a,b\}]} \tilde{x}_e = \sum_{e\in T_1[\{a,u\}]} \tilde{x}_e + \sum_{e\in T_1[\{u,b\}]} \tilde{x}_e$. If $a = u$, then $T_1[\{a,u\}]$ contains no edges, so $\sum_{e\in T_1[\{a,u\}]} \tilde{x}_e = 0$. If $a \neq u$ but $T_1[\{a,u\}] \cap E = \emptyset$, then $T_1[\{a,u\}] \subseteq D(e)$, where e is the edge in $T_1[\{a,u\}]$ incident to u, so $\sum_{e\in T_1[\{a,u\}]} \tilde{x}_e \leq w(e) < 1/4$. An analogous argument shows that $\sum_{e\in T_1[\{u,b\}]} \tilde{x}_e < 1/4$. This proves that $\sum_{e\in T_1[\{a,b\}]} \tilde{x}_e = \sum_{e\in T_1[\{a,u\}]} \tilde{x}_e + \sum_{e\in T_1[\{u,b\}]} \tilde{x}_e < 1/2$ for any two leaves $a, b \in X$ such that $T_1[\{a,b\}] \cap E = \emptyset$.

For any quartet $Q = \{a,b,c,d\} \in \mathcal{Q}$, we have $\sum_{e\in\mathcal{L}(Q)} \tilde{x}_e \geq 1$ because $\tilde{x}$ is a feasible fractional solution of (1). Thus, if $T_1|_Q \cong ab|cd$, then $\sum_{e\in T_1[\{a,b\}]} \tilde{x}_e \geq 1/2$ or $\sum_{e\in T_1[\{c,d\}]} \tilde{x}_e \geq 1/2$. Therefore, $T_1[\{a,b\}] \cap E \neq \emptyset$ or $T_1[\{a,b\}] \cap E \neq \emptyset$, that is, $\mathcal{L}(Q) \cap E \neq \emptyset$. Since this is true for every quartet $Q \in \mathcal{Q}$, E is a feasible solution of (1). □

Since the set of incompatible quartets $\mathcal{Q}$ in (1) contains at most $\binom{n}{4} = O(n^4)$ quartets, the LP relaxation of (1) has polynomial size. Thus, it can be solved in polynomial time using the ellipsoid algorithm [10] or any one of a number of more recent interior point methods.

The set E is then easily constructed in linear time: We intialize $E = \emptyset$ and traverse T_1 from the leaves towards the root. When visiting each edge $e = (u_e, v_e) \in T_1$, we compute $w(e)$ by summing the weights $w(f)$ of all edges $f = (u_f, v_f)$ with $u_f = v_e$ and adding $\tilde{x}_e$ to this sum. If $w(e) < 1/4$, then we take no further action for e. If $w(e) \geq 1/4$, then we add e to E and set $w(e) = 0$.

Together with Lemmas 4.1 and 4.2, this shows the following theorem.

Theorem 4.3. *There exists a polynomial-time 4-approximation algorithm for computing the MAF of a set of unrooted binary trees based on LP rounding.*

5. A family of tight inputs

Next we prove that the integrality gap of (1) is $4 - o(1)$ even for two input trees. This implies that no approximation algorithm that uses an optimal fractional solution of (1) (or any dual solution; see Section 6 for why this is important) as a lower bound on the optimal solution can achieve an approximation ratio of $4 - \varepsilon$, for any $\varepsilon > 0$.

Lemma 5.1. *There exists a fractional feasible solution $\tilde{x}$ of* (1) *that satisfies $\sum_{e\in E(T_1)} \tilde{x}_e = n/4$, where $n = |X|$.*

Proof. For every leaf $v \in X$, let e_v be the unique edge incident to v. Now consider the fractional solution $\tilde{x}$ that sets $\tilde{x}_{e_v} = 1/4$ for all $v \in X$, and $\tilde{x}_e = 0$ for any other edge. Clearly, $\sum_{e\in E(T_1)} \tilde{x}_e = n/4$.

To see that $\tilde{x}$ is feasible, observe that for every quartet $Q = \{a,b,c,d\} \in \mathcal{Q}$, we have $\{e_a, e_b, e_c, e_d\} \subseteq \mathcal{L}(Q)$. Since $\tilde{x}_{e_v} = 1/4$ for every leaf $v \in X$, this shows that $\sum_{e\in\mathcal{L}(Q)} \tilde{x}_e \geq \tilde{x}_{e_a} + \tilde{x}_{e_b} + \tilde{x}_{e_c} + \tilde{x}_{e_d} = 1$. Since this is true for every quartet $Q \in \mathcal{Q}$, $\tilde{x}$ is feasible. □

Lemma 5.2. *There exists an infinite family of pairs of trees such that any agreement forest of any pair (T_1, T_2) in this family has $n - o(n)$ components, where n is the size of the label set X of T_1 and T_2.*

Proof. Let $\ell \geq 4$ be an integer. Consider two trees T_1 and T_2 with leaf set $X = \{(i,j) \mid 1 \leq i, j \leq \ell\}$. Thus, $n = |X| = \ell^2$, that is, $\ell = \sqrt{n}$. Both T_1 and T_2 are caterpillars, that is, the internal vertices of each tree form a path, which we call the *spine* of the caterpillar. In T_1, the leaves are attached to this path sorted by their i-components and then by their j-components. In T_2, they are sorted by their j-components and then by their i-components. For an example, see Fig. 1.

Now we call an edge $e \in E(T_1)$ *separating* if there exists an index $1 \leq i < \ell$ such that e belongs to the path from (i,j) to $(i+1,j')$, for all $1 \leq j, j' \leq \ell$. Similarly, we call an edge $e \in E(T_2)$ separating if there exists an index $1 \leq j < \ell$ such that e belongs to the path from (i,j) to $(i', j+1)$, for all $1 \leq i, i' \leq \ell$. The separating edges are shown dotted in Fig. 1. Note that there are $\ell - 1$ separating edges in T_1, and $\ell - 1$ separating edges in T_2, $2\ell - 2$ separating edges in total.

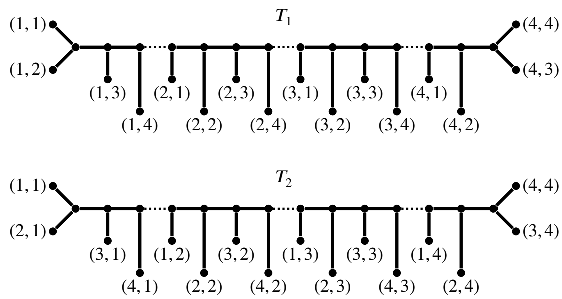

Fig. 1. The two trees T_1 and T_2 in the proof of Lemma 5.2 for $\ell = 4$.

Now assume that $\mathcal{F} = \{Y_1, \dots, Y_k\}$ is an AF of T_1 and T_2. For each component Y_h, let $S^1_{Y_h}$ be the set of separating edges of T_1 contained in $T_1[Y_h]$, let $S^2_{Y_h}$ be the set of separating edges of T_2 contained in $T_2[Y_h]$, and let $S_{Y_h} = S^1_{Y_h} \cup S^2_{Y_h}$. Since no two components of $\mathcal{F}$ overlap in either T_1 or T_2, we have $S_{Y_h} \cap S_{Y_{h'}} = \emptyset$, for any two distinct components $Y_h, Y_{h'} \in \mathcal{F}$, that is, $\sum_{h=1}^{k} |S_{Y_h}| \le 2\ell - 2$. Next we prove that $|Y_h| \le |S_{Y_h}| + 1$, for all $Y_h \in \mathcal{F}$. This implies that $n = \sum_{h=1}^{k} |Y_h| \le \sum_{h=1}^{k} (|S_{Y_h}| + 1) = \sum_{h=1}^{k} |S_{Y_h}| + k \le 2\ell - 2 + k$, so $k \ge n - 2\ell + 2 = n - 2\sqrt{n} + 2 = n - o(n)$, and the lemma follows.

Consider a component $Y_h = \{x_1, \dots, x_s\}$ of $\mathcal{F}$; let $x_r = (i_r, j_r)$, for all $1 \le r \le s$; and assume that the leaves in Y_h are indexed in the order in which they occur along T_1. For each index $1 \le r < s$, we choose a separating edge e_r of T_1 or T_2 such that $e_r \in S_{Y_h}$ and, for all $1 \le r_1 < r_2 < s$, $e_{r_1} \ne e_{r_2}$. This immediately implies that $|Y_h| = s \le |S_{y_h}| + 1$.

If $i_r < i_{r+1}$, then we choose e_r to be the i_rth separating edge in T_1. Otherwise, we must have $i_r = i_{r+1}$ and $j_r < j_{r+1}$, and we choose e_r to be the j_rth separating edge in T_2. Since the leaves in Y_h are indexed in the order in which they occur along T_1, we have $i_{r_1} \ne i_{r_2}$ for any two indices $r_1 \ne r_2$ such that $i_{r_1} < i_{r_1+1}$ and $i_{r_2} < i_{r_2+1}$. Thus, the separating edges chosen from T_1 are all distinct.

Next assume that there exist two indices $r_1 < r_2$ such that e_{r_1} and e_{r_2} are the same separating edge from T_2. Then $i_{r_1} = i_{r_1+1}$, $j_{r_1} < j_{r_1+1}$, $i_{r_2} = i_{r_2+1}$, $j_{r_2} < j_{r_2+1}$, and $j_{r_1} = j_{r_2}$. Since $j_{r_1} = j_{r_2}$, we must have $i_{r_1} \ne i_{r_2}$ and, therefore, $i_{r_1+1} \ne i_{r_2+1}$. Thus, $Q = \{x_{r_1}, x_{r_1+1}, x_{r_2}, x_{r_2+1}\} \subseteq Y_h$ is a quartet that satisfies $T_1|_Q \cong x_{r_1}x_{r_1+1}|x_{r_2}x_{r_2+1}$ and $T_2|_Q \cong x_{r_1}x_{r_2}|x_{r_1+1}x_{r_2+1}$, a contradiction because $\mathcal{F}$ is an AF of T_1 and T_2 and Y_h is a component of $\mathcal{F}$. Thus, all separating edges chosen from T_2 are also distinct. This finishes the proof. $\square$

Lemmas 5.1 and 5.2 together prove the following theorem.

Theorem 5.3. *The integrality gap of* (1) *is at least* $4 - o(1)$ *even if* $|\mathcal{T}| = 2$.

Proof. Consider any pair in the family of tree pairs provided by Lemma 5.2. The optimal integral solution of (1) for this pair of trees has objective function value $n - o(n)$. By Lemma 5.1, the optimal fractional solution has objective function value at most $n/4$. Thus, the integrality gap is at least $\frac{n-o(n)}{n/4} = 4 - o(1)$. $\square$

6. Conclusions

Theorems 4.3 and 5.3 are significant for at least three reasons.

First, Theorem 4.3 matches the approximation ratio of the significantly more complex 4-approximation algorithm of [3], which is based on purely combinatorial arguments though.

Second, in [6], (1) and the ILP version of its dual were used to prove that there exists a kernel of size $O(k \lg k)$ for the 2-state maximum parsimony distance of two trees. This dual is

$$
\begin{aligned}
&\text{Maximize} \sum_{Q \in \mathcal{Q}} y_Q \\
\text{s.t.} \quad &\sum_{e \in \mathcal{L}(Q)} y_Q \le 1 \qquad \forall e \in E(T_1) \\
&y_Q \in \{0, 1\} \qquad \forall Q \in \mathcal{Q}.
\end{aligned}
\tag{2}
$$

The key to bounding the size of the kernel was to prove that the gap between optimal integral solutions of (1) and (2) is at most $O(\lg k)$. If both (1) and (2) have a constant integrality gap, then this proves that the kernel in [6] is in fact a linear kernel for the 2-state maximum parsimony distance. Theorem 4.3 proves the first half of this conjecture.

Finally, our hope was to use primal-dual arguments based on (1) and (2) to obtain a 2-approximation algorithm, or at least a c-approximation algorithm with $c < 3$, for the TBR distance of two unrooted binary trees. Theorem 5.3 proves that this is impossible. Thus, if there exists a 2-approximation algorithm for the TBR distance of two unrooted binary trees, it needs to be based on a different ILP formulation, possibly one mimicking the ILP in [12], which was used to prove that the algorithm in [12] outputs a 2-approximation of a MAF for two *rooted* trees.

An interesting question for future work is whether the approach in this paper can also be used to easily compute a 3-approximation (or better) of a MAF of multiple *rooted* trees. Similar to the unrooted case, it *is* known how to compute a 3-approximation of a MAF of a set of rooted binary trees [3], but the algorithm is much more complicated than the algorithm presented in this paper. Extending our result to rooted trees appears to be non-trivial. As discussed in Section 2, a MAF cannot contain incompatible quartets (or triplets in the rooted case), and no two components of the MAF can overlap in any of the input trees. The key observation by Van Wersch et al. [13] was that destroying each incompatible quartet Q by cutting at least one edge in $\mathcal{L}(Q)$ also ensures that components of the resulting forest do not overlap in any of the input trees. This is the key to the simplicity of the ILP formulation (1), which references only one of the input trees explicitly and represents the necessary information about the remaining trees only as a global set $\mathcal{Q}$ of incompatible quartets, making no distinction for which pairs of trees a quartet in $\mathcal{Q}$ is an incompatible quartet. This simplicity of the ILP in turn seems to be the key to the simplicity of our algorithm, because the rounding procedure can be implemented by a simple traversal of the one tree explicitly encoded in the ILP. We tried to identify a similar set $\mathcal{L}(R) \subseteq E(T_1)$ for every incompatible triplet R of a set $\mathcal{T} \supseteq \{T_1\}$ of rooted trees, so that $T_1 - E$ is an AF of $\mathcal{T}$ if and only if it hits $\mathcal{L}(R)$ for every incompatible triplet R, but we were unsuccessful.

CRediT authorship contribution statement

Jordan Dempsey: Writing – review & editing, Writing – original draft, Investigation, Formal analysis. **Leo van Iersel:** Writing – review & editing, Resources, Investigation, Funding acquisition, Formal analysis. **Mark Jones:** Writing – review & editing, Validation, Investigation, Funding acquisition, Formal analysis. **Norbert Zeh:** Writing – review & editing, Writing – original draft, Investigation, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] B.L. Allen, M. Steel, Subtree transfer operations and their induced metrics on evolutionary trees, Ann. Comb. 5 (1) (Jun 2001) 1–15.
[2] J. Chen, J.-H. Fan, S.-H. Sze, Parameterized and approximation algorithms for the MAF problem in multifurcating trees, in: Proceedings of the 39th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2013), Springer, 2013, pp. 152–164.
[3] J. Chen, F. Shi, J. Wang, Approximating maximum agreement forest on multiple binary trees, Algorithmica 76 (2016) 867–889.
[4] H. Colonius, H.-H. Schulze, Trees Constructed from Empirical Relations, Inst. für Psychologie d. Techn. Univ., 1977.
[5] H. Colonius, H.-H. Schulze, Tree structures for proximity data, Br. J. Math. Stat. Psychol. 34 (2) (1981) 167–180.
[6] E. Deen, L. van Iersel, R. Janssen, M. Jones, Y. Murakami, N. Zeh, A near-linear kernel for bounded-state parsimony distance, J. Comput. Syst. Sci. 140 (2024) 103477.
[7] M. Hallett, C. McCartin, A faster FPT algorithm for the maximum agreement forest problem, Theory Comput. Syst. 41 (3) (2007) 539–550.
[8] J. Hein, T. Jiang, L. Wang, K. Zhang, On the complexity of comparing evolutionary trees, Discrete Appl. Math. 71 (1–3) (1996) 153–169.
[9] S. Kelk, S. Linz, R. Meuwese, Deep kernelization for the tree bisection and reconnection (tbr) distance in phylogenetics, J. Comput. Syst. Sci. 142 (2024) 103519.
[10] L.G. Khachiyan, A polynomial algorithm for linear programming, Dokl. Akad. Nauk USSR 244 (1979) 1093–1096, in Russian.
[11] S. Linz, C. Semple, A cluster reduction for computing the subtree distance between phylogenies, Ann. Comb. 15 (3) (Sept. 2011) 465–484.
[12] N. Olver, F. Schalekamp, S. van der Ster, L. Stougie, A. van Zuylen, A duality based 2-approximation algorithm for maximum agreement forest, Math. Program. 198 (1) (2023) 811–853.
[13] R. van Wersch, S. Kelk, S. Linz, G. Stamoulis, Reflections on kernelizing and computing unrooted agreement forests, Ann. Oper. Res. 309 (1) (2022) 425–451.
[14] C. Whidden, R.G. Beiko, N. Zeh, Fixed-parameter algorithms for maximum agreement forests, SIAM J. Comput. 42 (4) (2013) 1431–1466.
[15] C. Whidden, N. Zeh, A unifying view on approximation and FPT of agreement forests, in: Proceedings of the 9th International Workshop on Algorithms in Bioinformatics (WABI 2009), in: Lecture Notes in Computer Science, vol. 5724, Springer, 2009, pp. 390–402.