Fitting Distances by Tree Metrics Minimizing the Total Error within a Constant Factor

VINCENT COHEN-ADDAD, Google Research, Switzerland
DEBARATI DAS and EVANGELOS KIPOURIDIS, Department of Computer Science, University of Copenhagen, Denmark
NIKOS PAROTSIDIS, Google Research, Switzerland
MIKKEL THORUP, Department of Computer Science, University of Copenhagen, Denmark

We consider the numerical taxonomy problem of fitting a positive distance function $\mathcal{D} : \binom{S}{2} \rightarrow \mathbb{R}_{>0}$ by a tree metric. We want a tree T with positive edge weights and including S among the vertices so that their distances in T match those in $\mathcal{D}$. A nice application is in evolutionary biology where the tree T aims to approximate thebranching process leading to the observed distances in $\mathcal{D}$ [Cavalli-Sforza and Edwards 1967]. We consider the total error, that is, the sum of distance errors over all pairs of points. We present a deterministic polynomial time algorithm minimizing the total error within a constant factor. We can do this both for general trees and for the special case of ultrametrics with a root having the same distance to all vertices in S.

The problems are APX-hard, so a constant factor is the best we can hope for in polynomial time. The best previous approximation factor was $O((\log n)(\log\log n))$ by Ailon and Charikar [2005], who wrote "determining whether an $O(1)$ approximation can be obtained is a fascinating question."

CCS Concepts: • **Theory of computation → Approximation algorithms analysis;**

Additional Key Words and Phrases: Approximation algorithms, phylogenic reconstructions, hierarchical clustering, tree metrics, ultrametrics

ACM Reference Format:
Vincent Cohen-Addad, Debarati Das, Evangelos Kipouridis, Nikos Parotsidis, and Mikkel Thorup. 2024. Fitting Distances by Tree Metrics Minimizing the Total Error within a Constant Factor. *J. ACM* 71, 2, Article 10 (April 2024), 41 pages. https://doi.org/10.1145/3639453

Research of Debarati Das, Evangelos Kipouridis and Mikkel Thorup is supported by VILLUM Investigator Grant 16582, Basic Algorithms Research Copenhagen (BARC), Denmark. Evangelos Kipouridis has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 801199.
Authors' addresses: V. Cohen-Addad and N. Parotsidis, Google Research, Switzerland; e-mails: cohenaddad@google.com, nikosp@google.com; D. Das, E. Kipouridis, and M. Thorup, Department of Computer Science, University of Copenhagen, Denmark; e-mails: debaratix710@gmail.com, kipouridis@di.ku.dk, mikkel2thorup@gmail.com.

ACM 0004-5411/2024/04-ART10
https://doi.org/10.1145/3639453

1 INTRODUCTION

Taxonomy or hierarchical classification of species goes back at least to discussions between Aristotle and his teacher Plato[1] (~350 BC), while modern taxonomy is often attributed to Linnaeus[2] (~1750). The discussions of evolution in the 19th century clarified the notion of evolutionary trees, or phylogenies, and the notion that species were close due to a common past ancestor. Such evolutionary trees are seen in the works of Hitchcock[3] (1840) and Darwin[4] (1859). Viewing the descendants of each node as a class, the evolutionary tree induces a hierarchical classification.

In the 1960s came the interest in computing evolutionary trees based on present data, the so-called numerical taxonomy problem [11, 49, 50]. Our focus is on the following simple model by Cavalli-Sforza and Edwards from 1967 [11]. In an evolutionary tree, let the edge between the child and its parent be weighted by the evolutionary distance between them. Then the evolutionary distance between any two species is the sum of weights on the unique simple path between them. We note that the selection of the root plays no role for the distances. What we are saying is that any tree with edge weights induces distances between its nodes, a so-called tree metric assuming that all weights are positive.

We now have the converse reconstruction problem of numerical taxonomy [11, 49, 50]: Given a set S of species with measured distances between them, find a tree metric matching those observed distances on S. Thus we are looking for an edge-weighted tree T that includes S among its nodes with the right distances between them. Importantly, T may have nodes not in S representing ancestors explaining proximity between different species. The better the tree metric T matches the measured distances on S, the better the tree T explains these measured distances.

Other applications. This very basic reconstruction problem also arises in various other contexts. First, concerning the evolutionary model, it may be considered too simplistic to just add up distances along the paths in the tree. Some changes from parent to child could be reverted for a grandchild. Biologists [12, 35] have suggested stochastic models for probabilistic changes that also have a chance of being reverted further down. However, Farach and Kannan [33] have shown that applying logarithms appropriately, we can convert estimated distances into some other distances for which we find a matching tree metric that we can then convert back into a maximum likelihood stochastic tree. The basic point is that finding tree metrics can be used as powerful tool to invert evolution even in cases where tree metric model does not apply directly.

Obviously, the numerical taxonomy problem is equally relevant to other historical sciences with an evolutionary branching process leading to evolutionary distances, e.g., historical linguistics.

More generally, if we can approximate distances with a tree metric, then the tree of this metric provides a very compact and convenient representation that is much easier to navigate than a general distance function. Picking any root, the tree induces a distance-based hierarchical classification, and referring to the discussions between Plato and Aristotle's, humans have been interested in such hierarchical classifications since ancient times.

It is not just humans but also computers that understand trees and tree metrics much better than general metrics. Many questions that are NP-hard to answer in general can be answered very efficiently based on trees (see, e.g., Chapter 10.2, "Solving NP-Hard Problems on Trees," in Reference [42]).

Computing "good" tree representations is nowadays also a major tool to learn from data. In this context, we are sometimes interested in a special kind of tree metrics, called *ultrametrics*, defined

[1]https://iep.utm.edu/classifi/,InternetEncyclopediaofPhilosophy
[2]https://britannica.com/science/taxonomy/The-Linnaean-system
[3]https://en.wikipedia.org/wiki/Edward_Hitchcock
[4]https://en.wikipedia.org/wiki/On_the_Origin_of_Species

by rooted trees whose sets of leaves is S and where the leaf-to-root distance is the same for all points in S. Equivalently, an ultrametric is a metric so that for any three points i, j, k, the distance from i to j is no bigger than the maximum of the distance from i to j and the distance from j to k.[5]

An ultrametric can be seen as modeling evolution that is linear over time. This may not be the case in biology, where the speed of evolution depends on the local evolutionary pressure for example. However, ultrametrics are key objects in machine learning and data analysis, see e.g., Reference [10], and there are various algorithms for embedding arbitrary metrics into ultrametrics such as the popular "linkage" algorithms (single, complete or average linkage), see also References [24, 45].

1.1 Tree Fitting (Numerical Taxonomy Problem)

Typically our measured distances do not have an exact fit with any tree metric. We then have the following generic optimization problem for any L_p-norm:

Problem: L_p-fitting tree (ultra) metrics.

Input: A set S with a distance function $\mathcal{D} : \binom{S}{2} \to \mathbb{R}_{>0}$.[6]

Desired Output: A tree metric (or ultrametric) T that spans S and fits $\mathcal{D}$ in the sense of minimizing the L_p-norm

$$\|T - \mathcal{D}\|_p = \left(\sum_{\{i,j\} \in \binom{S}{2}} |\mathrm{dist}_T(i,j) - \mathcal{D}(i,j)|^p \right)^{1/p}. \tag{1}$$

Cavalli-Sforza and Edwards [11] introduced this numerical taxonomy problem for both tree and ultrametrics in the L_2-norm in 1967. Farris suggested using L_1-norm in 1972 [35, p. 662].

1.2 Our Result

In this article, we focus on the L_1-norm, that is, the total sum of errors. The problem is APX-hard for both tree metrics and ultrametrics (see Section 9 and Reference [3]), so a constant approximation factor is the best we can hope for in polynomial time. The best previous approximation factor for both tree metrics and ultrametrics was $O((\log n)(\log\log n))$ by Ailon and Charikar [3].

In this article, we present a deterministic polynomial time constant factor approximation both for tree metrics and for ultrametrics, that is, in both cases, we can find a tree T minimizing the L_1-norm within a constant factor of the best possible.

Thus, we will prove the following theorem.

Theorem 1.1. *The L_1-fitting tree metrics problem can be solved in deterministic polynomial time within a constant approximation factor. The same holds for the L_1-fitting ultrametrics problem.*

1.3 History of L_p Tree Fitting

Since Cavalli-Sforza and Edwards introduced the tree fitting problem, the problem has collected an extensive literature. In 1977 [54], it was shown that if there is a tree metric coinciding exactly with $\mathcal{D}$, then it is unique and it can be found in time linear in the input size, i.e., $O(|S|^2)$ time. The same then also holds trivially for ultrametrics. Unfortunately there is typically no tree metric coinciding exactly with $\mathcal{D}$, and in 1987 [28] it was shown that for L_1 and L_2 the numerical taxonomy problem is NP-hard in the tree metric and the ultrametric cases. The problems are in fact APX-hard (see Section 9), which rules out the possibility of a polynomial-time approximation scheme. Thus, a

[5] https://en.wikipedia.org/wiki/Ultrametric_space
[6] $\binom{S}{k}$ denotes all subsets of S of size k.

constant factor, like ours for L_1, is the best one can hope for from a complexity perspective for these problems.

For the L_∞ numerical taxonomy problem, there was much more progress. In 1993 [34], it was shown that for the ultrametric case an optimal solution can be found in $\Theta(|S|^2)$ time. More recently, it was shown that when the points are embedded into $\mathbb{R}^d$ and the distances are given by the pairwise Euclidean distances, the problem can be approximated in subquadratic time [22, 25]. For the general trees case (still in the L_∞-norm objective), Reference [2] gave an $O(|S|^2)$ algorithm that produces a constant factor approximation and proved that the problem is APX-hard (unlike the ultrametric case).

The technical result from Reference [2] was a general reduction from general tree metrics to ultrametrics. It modifies the input distance matrix and asks for fitting this new input with an ultrametric that can later be converted to a tree metric for the original distance matrix. The result states that for any p, if we can minimize the restricted ultrametric L_p error within a factor α in polynomial-time, then there is a polynomial-time algorithm that minimizes the tree metric L_p error within a factor 3α. The reduction from Reference [2] imposes a certain restriction on the ultrametric, but the restriction is not problematic, and in Section 8, we will even argue that the restriction can be completely eliminated with a slightly modified reduction. With n species, the reduction from tree metrics to ultrametrics can be performed in time $O(n^2)$. Applying this to the optimal ultrametric algorithm from Reference [34] for the L_∞-norm objective yielded a factor 3 for general metrics for the L_∞-norm objective. The generic impact is that *for any L_p, later algorithms only had to focus on the ultrametric case to immediately get results for the often more important tree metrics case, up to losing a factor 3 in the approximation guarantee.* Indeed, the technical result of this article is a constant factor approximation for ultrametric. Thus, letting TreeMetric and UltraMetric respectively denote the approximation factors of the tree metric and ultrametric problems, we have

$$\text{TreeMetric} \leq 3 \cdot \text{UltraMetric}$$

For L_p-norms with constant p, the developments have been much slower. Ma et al. [44] considered the problem of finding the best L_p fit by an ultrametric where distances in the ultrametric are no smaller than the input distances. For this problem, they obtained an $O(n^{1/p})$ approximation.

Later, Dhamdhere [29] considered the problem of finding a line metric to minimize additive distortion from the given data (measured by the L_1-norm) and obtained an $O(\log n)$ approximation. In fact, his motivation for considering this problem was to develop techniques that might be useful for finding the closest tree metric with distance measured by the L_1-norm. Harb, Kannan, and McGregor [40] developed a factor $O(\min\{n, k \log n\}^{1/p})$ approximation for the closest ultrametric under the L_p-norm where k is the number of distinct distances in the input.

The best bounds known for the ultrametric variant of the problem are due to Ailon and Charikar [3]. They first focus on ultrametrics in L_1 and show that if the distance matrix has only k distinct distances, then it is possible to approximate the L_1 error within a factor $k + 2$. Next they obtain an LP-based $O((\log n)(\log\log n))$ approximation for arbitrary distances matrices. Finally, they sketch how it can be generalized to an $O(((\log n)(\log\log n))^{1/p})$ approximation of the L_p error for any p. Using the reduction from Reference [2], they also get an $O(((\log n)(\log\log n))^{1/p})$ approximation for tree metrics under the L_p-norm objective. The $O(((\log n)(\log\log n))^{1/p})$ approximation comes from an $O((\log n)(\log\log n))$ approximation of the pth moment of the following function:

$$\|T - \mathcal{D}\|_p^p = \left(\sum_{\{i,j\} \in \binom{S}{2}} |\text{dist}_T(i,j) - \mathcal{D}(i,j)|^p \right). \tag{2}$$

Table 1. Tree Fitting Approximation Factors

Norm	L_1	$L_p, p < \infty$	L_∞
Treemetric	$\Theta(1)$	$O(((\log n)(\log\log n))^{1/p})$	$\Theta(1)$
Ultrametric	$\Theta(1)$	$O(((\log n)(\log\log n))^{1/p})$	1

Technically, Ailon and Charikar [3] present a simple LP relaxation for L_1 ultrametric fitting—an LP that will also be used in our article. They get their $O((\log n)(\log\log n))$ approximation using an LP rounding akin to the classic $O(\log n)$ rounding of Leighton and Rao for multicut [43]. The challenge is to generalize the approach to deal with the hierarchical issues associated with ultrametric and show that this can be done paying only an extra factor $O(\log\log n)$ in the approximation factor. Next they show that their LP formulation and rounding is general enough to handle different variants, including other L_p-norms as mentioned above, but also they can handle the *weighted case*, where for each pair of species i, j, the error contribution to the overall error is multiplied by a value w_{ij}. However, this weighted problem captures the multicut problem (and the weighted minimization correlation clustering problem) [3]. Since the multicut cannot be approximated within a constant factor assuming the unique games conjecture [19] and the best-known approximation bound remains $O(\log n)$, it is beyond reach of current techniques to do much better in these more general settings.

Ailon and Charikar [3] conclude that "determining whether an $O(1)$ approximation can be obtained is a fascinating question. The LP formulation used in our [their] work could eventually lead to such a result." For their main LP formulation for the (unweighted) L_1 ultrametric fitting, the integrality gap was only known to be somewhere between 2 and $O((\log n)(\log\log n))$. To break the $\log n$-barrier, we must come up with a radically different way of rounding this LP and free ourselves from the multicut-inspired approach.

For L_1 ultrametric fitting, we give the first constant factor approximation, and we show this can be obtained by rounding the LP proposed by Ailon and Charikar, thus demonstrating a constant integrality gap for their LP. Our solution breaks the $\log n$ barrier using the special combinatorial structure of the L_1 problem.

Stepping a bit back, having different solutions for different norms should not come as a surprise. As an analogue, take the generic problem of placing k facilities in such a way that each of n cities is close to the nearest facility. Minimizing the vector of distances in the L_1-norm is called the k-median problem. In the L_2-norm, it is called the k-means problem and in the L_∞-norm the k-center problem. Indeed, while the complexity of the k-center problem has been settled in the mid-1980s thanks to Gonzalez's algorithm [38], it remained a major open problem for the next 15 years to obtain constant factor approximation for the k-median and the k-means problems. Similarly, our understanding of the k-means problem (L_2-objective) remains poorer than our understanding of the k-median problem, and the problem is in fact provably harder (no better than $1+8/e$-approximation algorithm [39], while k-median can be approximated within a factor 2.675 [9]).

For our tree fitting problem, the L_∞-norm has been understood since the 1990s, and our result shows that the L_1-norm admits a constant factor approximation algorithm. The current status of affairs for tree and ultrametrics is summarized in Table 1. The status for L_p tree fitting is that we have a good constant factor approximation if we want to minimize the total error L_1 or the maximal error L_∞. For all other L_p-norms, we only have the much weaker but general $O(((\log n)(\log\log n))^{1/p})$ approximation from Reference [3]. In particular, we do not know if anything better is possible with L_2. The difference is so big that even if we are in a situation where we would normally prefer an L_2 approximation, our much better approximation guarantee with L_1 might be preferable.

1.4 Other Related Work

Computational Biology. Researchers have also studied reconstruction of phylogenies under stochastic models of evolution (see Farach and Kannan [33] or Mossel et al. [46] and the references therein; see also Henzinger et al. [41]).

Finally, related to the hierarchical correlation clustering problem that we introduce in this article is the hierarchical clustering problem introduced by Dasgupta [27] where the goal is, given a similarity matrix, to build a hierarchical clustering tree where the more similar two points are, the lower in the tree they are separated (formally, a pair (u, v) induces a cost of similarity(u, v) times the size of the minimal subtree containing both u and v, the goal is to minimize the sum of the costs of the pairs). This has received a lot of attention in the past few years (References [5, 14–16, 18, 23, 24, 45, 47], see also References [1, 6, 13, 21]) but differs from our settings, since the resulting tree may not induce a metric space.

Metric Embeddings. There is a large body of work of metric embedding problems. For example, the metric violation distance problem asks to embed an arbitrary distance matrix into a metric space while minimizing the L_0-objective (i.e., minimizing the number of distances that are not preserved in the metric space). The problem is considerably harder and is only known to admit an $O(\mathrm{OPT}^{1/3})$-approximation algorithm [31, 32, 36], while no better than a 2 hardness of approximation is known. More practical results on this line of work includes References [51] and [37]. Sidiropoulos et al. [48] also considered the problem of embedding into metric, ultrametric, and so on, while minimizing the total number of outlier points.

There is also a rich literature on metric embedding problems where the measure of interest is the multiplicative distortion, and the goal of the problem is to approximate the absolute distortion of the metric space (as opposed to approximating the optimal embedding of the metric space). Several such problems have been studied in the context of approximating metric spaces via tree metrics (e.g., References [8, 30]). The objective of these works is very different, since they are focused on the absolute expected multiplicative distortion over all input metrics while we aim at approximating the *optimal* expected additive distortion for each individual input metric.

While the embedding techniques developed in References [8, 30] and related works have been very successful for designing approximation algorithms for various problems in a variety of contexts, they are not aimed at numerical taxonomy. Their goal is to do something for general metrics. However, for our tree-fitting problem, the idea is that the ground truth is a tree, e.g., a phylogeny, and that the distances measured, despite noise and imperfection of the model, are close to the metric of the true tree. To recover an approximation to the true tree, we therefore seek a tree that compares well against the best possible fit of a tree metric.

1.5 Techniques

We will now discuss the main idea of our algorithm. Our solution will move through several combinatorial problems that code different aspects of the L_1-fitting of ultrametrics but that do not generalize nicely to other norms.

Our result follows from a sequence of constant-factor-approximation reductions between problems. To achieve our final goal, we introduce several new problems that have a key role in the sequence of reductions. Some of the reductions and approximation bounds have already been extensively studied (e.g., Correlation Clustering). A roadmap of this sequence of results is given in Figure 1. While some the reductions and techniques we use for our new constant factor approximation were known, we do not make use of anything published since the previous $O((\log n)(\log\log n))$ factor approximation of Ailon and Charikar [4].

$$\begin{aligned}
\text{TreeMetric} &\le (3 + o(1)) \cdot \text{UltraMetric}, && \text{(A)}\\
\text{UltraMetric} &\le \text{HierCorrClust}, && \text{(B)}\\
\text{HierCorrClust} &\le (\text{CorrClust} + 1)(\text{HierClustAgree} + 1) - 1, && \text{(C)}\\
\text{CorrClust} &= O(1), && \text{(D)}\\
\text{HierClustAgree} &= O(1). && \text{(E)}
\end{aligned}$$

Approximation factors. Abbreviated problem names used as approximation factors.

```
1:  procedure TreeMetric($S, \mathcal{D}$)                              ▷ See Section 8
2:      Reduction to UltraMetric based on Reference [2]
3:
4:  procedure UltraMetric($S, \mathcal{D}$)                             ▷ See Section 7
5:      Reduction to HierCorrClust based on Reference [3, 40]
6:
7:  procedure HierCorrClust($S, E^{(*)}, \delta^{(*)}$)                ▷ NEW. See Section 3
8:      for $t \in [\ell]$ do $Q^{(t)} \leftarrow$ CorrClust($E^{(t)}$)
9:      return HierClustAgree($S, Q^{(*)}, \delta^{(*)}$)
10:
11: procedure CorrClust($S, E$)                                         ▷ See Section 1.5.1
12:     Use Algorithm from Reference [7]
13:
14: procedure HierClustAgree($S, Q^{(*)}, \delta^{(*)}$)                ▷ NEW. See Section 4
15:     $x^{(*)} \leftarrow$ Solve(LP-relaxation($S, Q^{(*)}, \delta^{(*)}$))
16:     $L^{(*)} \leftarrow$ LP-Cleaning($S, Q^{(*)}, x^{(*)}$)
17:     return Derive-Hierarchy($S, L^{(*)}$)
```

Fig. 1. Roadmap leading to our result for L_1-fitting tree metrics.

1.5.1 *Correlation Clustering.*

Our algorithms will use a subroutine for what is known as the unweighted minimizing-disagreements correlation clustering problem on complete graphs [7]. We simply refer to this problem as Correlation Clustering throughout the article.

First, for any family P of disjoint subsets of S, let

$$\mathcal{E}(P) = \bigcup_{T \in P} \binom{T}{2}.$$

Thus $\mathcal{E}(P)$ represents the edge sets over an isolated clique over each set T in P. Often P will be a partition of S, that is, $\bigcup P = S$.

The *correlation clustering* takes as input an edge set $E \subseteq \binom{S}{2}$ and seeks a partition P minimizing

$$|E \triangle \mathcal{E}(P)|,$$

where $\triangle$ denotes symmetric difference. Bansal et al. [7] presented a deterministic constant factor approximation for this correlation clustering problem. Thus,

$$\text{CorrClust} = O(1). \qquad \text{(D) from Figure 1}$$

We note that better approximation factors have been found, e.g., a randomized polynomial time $1.994 + \epsilon$ factor approximation from Reference [26] (see also References [4, 20]) and a 2.5 deterministic approximation algorithm [52] are known. However, for our purposes, it suffices to know that a constant factor can be achieved in polynomial time.

It is well known that correlation clustering is equivalent to ultrametric fitting with two distances (see, e.g., Reference [40]). Also, we note that Ailon and Charikar, who presented the previous best $O((\log n)(\log\log n))$ approximation for tree metrics and ultrametrics at FOCS'05 [3], had presented a 2.5 approximation for correlation clustering at the preceding STOC'05 with Newman [4]. In fact, inspired by this connection, they proposed in Reference [3] a pivot-based $(M+2)$-approximation algorithm for the L_1 ultrametric problem where M is the number of distinct input distances. Thus, involving correlation clustering for fitting ultrametrics is not a new idea. The novelty is the way we use it so as to get the desired constant factor approximation for tree and ultrametrics.

1.5.2 Hierarchical Correlation Clustering. We are going to work with a generalization of the problem of L_1-fitting ultrametric that is implicit in previous work [3, 40]. In fact, Reference [3] contains an integer program capturing exactly this problem, and indeed it is the LP relaxation of this problem that they round within a factor of $O((\log n)(\log\log n))$. However, here we will exploit the generality of hierachical correlation clustering in new interesting ways.

Problem Hierarchical Correlation Clustering.
Input The input is ℓ weights $\delta^{(1)}, \ldots, \delta^{(\ell)} \in \mathbb{R}_{>0}$ and ℓ edge sets $E^{(1)}, \ldots, E^{(\ell)} \subseteq \binom{S}{2}$.
Desired output ℓ partitions $P^{(1)}, \ldots, P^{(\ell)}$ of S that are hierarchical in the sense that $P^{(t)}$ subdivides $P^{(t+1)}$ and such that we minimize

$$\sum_{t=1}^{\ell} \delta^{(t)} |E^{(t)} \triangle \mathcal{E}(P^{(t)})|. \tag{3}$$

Thus, we are having a combination of ℓ correlation clustering problems where we want the output partitions to form a hierarchy and where the objective is to minimize a weighted sum of the costs for each level problem.

We shall review the reduction from Reference [3] of L_1-fitting of ultrametrics to hierarchical correlation clustering in Section 7. The instances we get from ultrametrics will always satisfy $E^{(1)} \subseteq \cdots \subseteq E^{(\ell)}$, but as we shall see shortly, our new algorithms will reduce to instances where this is not the case, even if the original input is from an ultrametric.

1.5.3 Hierarchical Cluster Agreement. We will be particularly interested in the following special case of Hierarchical Correlation Clustering.

Problem Hierarchical Cluster Agreement.
Input The input is ℓ weights $\delta^{(1)}, \ldots, \delta^{(\ell)} \in \mathbb{R}_{>0}$ and ℓ partitions $Q^{(1)}, \ldots Q^{(\ell)}$ of S.
Desired output ℓ partitions $P^{(1)}, \ldots, P^{(\ell)}$ of S that are hierarchical in the sense that $P^{(t)}$ subdivides $P^{(t+1)}$ and such that we minimize

$$\sum_{t=1}^{\ell} \delta^{(t)} |\mathcal{E}(Q^{(t)}) \triangle \mathcal{E}(P^{(t)})|. \tag{4}$$

This is the special case of hierarchical correlation clustering, where the input edge set $E^{(t)}$ are the disjoint clique edges from $\mathcal{E}(Q^{(t)})$. The challenge is that the input partitions may disagree in the sense that $Q^{(t)}$ does not subdivide $Q^{(t+1)}$, or, equivalently, $\mathcal{E}(Q^{(t)}) \not\subseteq \mathcal{E}(Q^{(t+1)})$, so now we have to find the best hierarchical agreement.

We are not aware of any previous work on hierarchical cluster agreement, but it plays a central role in our hierarchical correlation clustering algorithm, outlined below.

1.6 High-level Algorithm for Hierarchical Correlation Clustering

Our main technical contribution in this article is solving Hierarchical Correlation Clustering. Reductions from Ultrametric to Hierarchical Correlation Clustering, and from general Tree Metric to Ultrametric, are already known from References [3, 40] and [2], respectively. We discuss both reductions in Sections 7 and 8. This includes removing some restrictions in the reduction from Tree Metrics to Ultrametrics.

Focusing on Hierarchical Correlation Clustering, our input is the ℓ weights $\delta^{(1)}, \dots, \delta^{(\ell)} \in \mathbb{R}_{>0}$ and ℓ edge sets $E^{(1)}, \dots, E^{(\ell)} \subseteq \binom{S}{2}$.

Step 1: Solve correlation clustering independently for each level. The first step in our solution is to solve the correlation clustering problem defined by $E^{(t)}$ for each level $t = 1, \dots, \ell$ independently, thus obtaining an intermediate partitioning Q_t. As we mentioned in Section 1.5.1, this can be done so that Q_t minimizes $|E^{(t)} \triangle E(Q_t)|$ within a constant factor.

Step 2: Solve hierarchical cluster agreement. We now use the ℓ weights $\delta^{(1)}, \dots, \delta^{(\ell)} \in \mathbb{R}_{>0}$ and ℓ partitions $Q^{(1)}, \dots, Q^{(\ell)}$ of S as input to the hierarchical cluster agreement problem, which we solve using an LP very similar to the one Ailon and Charikar [3] used to solve general hierarchical correlation clustering within a $O((\log n)(\log\log n))$ factor. However, when applied to the special case of hierarchical cluster agreement, it allows an unusual kind of constant-factor LP rounding. We use the LP variables (which only indicate how much pairs of species should be together) to decide which sets from the input partitions are important to the hierarchy and which sets can be ignored. The important sets may not yet be hierarchical (or laminar), but we show that some small modifications suffice. This is done by a relatively simple combinatorial algorithm that modify the important sets bottom-up to generate the hierarchical output partitions $P^{(1)}, \dots, P^{(\ell)}$. The result is a poly-time constant factor approximation for hierarchical cluster agreement, that is,

$$\text{HierClustAgree} = O(1).$$

The output partitions $P^{(1)}, \dots, P^{(\ell)}$ are also returned as output to the original hierarchical correlation clustering problem.

We now provide a high level overview and the intuition behind the hierarchical cluster agreement algorithm. The algorithm can be broadly divided into two parts.

LP cleaning. We start by optimally solving the LP based on the weights $\delta^{(1)}, \dots, \delta^{(\ell)}$ and partitions $Q^{(1)}, \dots, Q^{(\ell)}$. For each level $t \in \{1, \dots, \ell\}$, we naturally think of the relevant LP variables as distances and call them LP distances. That is because a small value means that the LP wants the corresponding species to be in the same part of the output partition at level t and vice versa, while the LP constraints also enforce the triangle inequality. The weights $\delta^{(1)}, \dots, \delta^{(\ell)}$ impact the optimal LP variables but will otherwise not be used in the rest of the algorithm.

Using the LP distances, we clean each set in every partition $Q^{(t)}$ independently. The objective of this step (LP-Cleaning - Algorithm 2) is to keep only the sets where most species are very close to each other and far away from the species not in the set. All other sets are disregarded. We process the sets of each level independently. The algorithm may only slightly modify sets that are not disregarded when processing their level. The property that we can clean each set independently to decide whether it is important or not, without looking at any other sets makes this part of our algorithm quite simple.

Omitting exact thresholds for simplicity, the algorithm works as follows. We process each set $C_I \in Q^{(t)}$ by keeping only those species that are at very small LP distance from at least half of the other species in C_I and at large LP distance to almost all the species outside C_I. Let us note that by triangle inequality and the pigeonhole principle, all species left in a set are at relatively small distance from each other. After this cleaning process, we only keep a set if at least 90% of its species are still intact, and we completely disregard it otherwise. The LP cleaning algorithm outputs the sequence $L^{(*)} = (L^{(1)}, \dots, L^{(\ell)})$, where $L^{(t)}$ is the family of surviving cleaned sets from $Q^{(t)}$.

Derive hierarchy. Taking $L^{(*)}$ as input, in the next step the algorithm Derive-Hierarchy (Algorithm 3) computes a hierarchical partition $P^{(*)} = (P^{(1)}, \ldots, P^{(\ell)})$ of S. This algorithm works bottom-up while initializing an auxiliary bottom most level of the hierarchy with $|S|$ sets where each set is a singleton and corresponds to a species of S. Then the algorithm performs ℓ iterations where at the tth iteration it processes all the disjoint sets in $L^{(t)}$ and computes partition $P^{(t)}$ while ensuring that at the end of the iteration $P^{(1)}, \ldots, P^{(t)}$ are hierarchical. An interesting feature of our algorithm is that while creating $P^{(t)}$ we do not make any changes to the lower partitions $P^{(1)}, \ldots, P^{(t-1)}$. Next, we discuss how to compute $P^{(t)}$ given $L^{(t)}$ and the lower partitions $P^{(1)}, \ldots, P^{(t-1)}$.

Consider a set $C_{LP} \in L^{(t)}$. Now if for each lower level set C' either $C' \cap C_{LP} = \emptyset$ or $C' \subseteq C_{LP}$, then introducing C_{LP} at level t does not violate the hierarchy property. Otherwise, let C' be a lower level set such that $C' \cap C_{LP} \neq \emptyset$ and $C' \nsubseteq C_{LP}$. Note that we already mentioned, once created, C' is never modified while processing upper level sets. Thus, to ensure the hierarchy condition, the algorithm can either extend C_{LP} so that it completely covers C' or can discard the common part from C_{LP}.

In the process of modifying C_{LP} (where we can add or discard some species from it), at any point we define the core of C_{LP} to be the part that comes from the set created initially. Now to resolve the conflict between C_{LP} and C', we work as follows. If the core of C_{LP} intersects the core of C', then we extend C_{LP} so that C' becomes a subset of it. Omitting technical details, there are two main ideas here: First, we ensure that the number of species in C' (respectively, C_{LP}) that are not part of its core is negligible with respect to the size of C' (respectively, C_{LP}). Furthermore, since the cores of C_{LP}, C' have at least one common species, using triangle inequality we can claim that any pair of species from the cores of C', C_{LP} also have small LP distance; therefore, nearly all pairs of species in C_{LP}, C' have small LP distance, meaning that the extension of C_{LP} is desirable (i.e., its cost is within a constant factor from the LP cost while it ensures the hierarchy).

Here we want to emphasize the point that because of the LP-cleaning, we can ensure that for any lower level set C' at level t there exists at most one set whose core has an intersection with the core of C'. We call this the *hierarchy-friendly* property of the LP cleaned sets. This property is crucial for consistency, as it ensures that at level t there cannot exist more than one sets that are allowed to contain C' as a subset.

In the other case, where the cores of C_{LP} and C' do not intersect, the algorithm removes $C_{LP} \cap C'$ from C_{LP}. The analysis of this part is more technical but follows the same arguments, namely using the aforementioned properties of LP-cleaning along with triangle inequality.

After processing all the sets in $L^{(t)}$, the algorithm naturally combines these processed sets with $P^{(t-1)}$ to generate $P^{(t)}$, thus ensuring that $P^{(1)}, \ldots, P^{(t)}$ are hierarchical.

High-level analysis. We will prove that the partitions $P^{(1)}, \ldots, P^{(\ell)}$ solves the original hierarchical clustering problem within a constant factor.

Using triangle inequality, we are going to show that the switch in Step 1, from the input edge sets $E^{(1)}, \ldots, E^{(\ell)}$ to the partitions $Q^{(1)}, \ldots, Q^{(\ell)}$ costs us no more than the approximation factor of correlation clustering used to generate each partition. This then becomes a multiplicative factor on our approximation factor for hierarchical cluster agreement, more specifically,

$$\text{HierCorrClust} < (\text{CorrClust} + 1)(\text{HierClustAgree} + 1).$$

We will even show that we can work with the LP from Reference [3] for the original hierarchical correlation clustering problem, and get a solution demonstrating a constant factor integrality gap.

1.7 Organization of the Article

In Section 2, we present the LP formulation and related definitions for Hierarchical Correlation Clustering. In Section 3, we show how to reduce Hierarchical Correlation Clustering to

Hierarchical Cluster Agreement. In Section 4, we present the algorithm for Hierarchical Cluster Agreement, and in Section 5 we analyze it. In Section 6, we show that the LP formulation for Hierarchical Correlation Clustering has constant integrality gap. In Section 7, we show how L_p-fitting ultrametrics reduces to Hierarchical Correlation Clustering. In Section 8, we discuss the reduction from L_p-fitting tree metrics to L_p-fitting ultrametrics. In Section 9, we prove APX-Hardness of L_1-fitting ultrametrics and L_1-fitting tree metrics. We conclude in Section 10.

2 LP DEFINITIONS FOR HIERARCHICAL CORRELATION CLUSTERING

In this section, we present the IP/LP formulation of Hierarchical Correlation Clustering, implicit in References [3, 40]. In what follows, we use $[n]$ to denote the set $\{1, \dots, n\}$.

Definition 2.1 (IP/LP Formulation of Hierarchical Correlation Clustering). Given is a set S, ℓ positive numbers $\delta^{(1)}, \dots, \delta^{(\ell)}$ and edge-sets $E^{(1)}, \dots, E^{(\ell)} \subseteq \binom{S}{2}$. The objective is

$$min \sum_{t=1}^{\ell} \delta^{(t)} \left(\sum_{\{i,j\} \in E^{(t)}} x_{i,j}^{(t)} + \sum_{\{i,j\} \notin E^{(t)}} (1 - x_{i,j}^{(t)}) \right),$$

subject to the constraints

$$x_{i,j}^{(t)} \leq x_{i,k}^{(t)} + x_{j,k}^{(t)} \qquad \forall \{i,j,k\} \in \binom{S}{3}, t \in [\ell], \tag{5}$$

$$x_{i,j}^{(t)} \geq x_{i,j}^{(t+1)} \qquad \forall \{i,j\} \in \binom{S}{2}, t \in [\ell - 1], \tag{6}$$

$$x_{i,j}^{(t)} \in \begin{cases} \{0,1\} & \text{if IP} \\ [0,1] & \text{if LP} \end{cases} \qquad \forall \{i,j\} \in \binom{S}{2}, t \in [\ell]. \tag{7}$$

Concerning the IP, the values $x_{i,j}^{(t)}$ encode the hierarchical partitions, with $x_{i,j}^{(t)} = 0$, meaning that i, j are in the same part of the partition at level t, and $x_{i,j}^{(t)} = 1$, meaning that they are not. Inequality (5) ensures that the property of being in the same part of a partition is transitive. Inequality (6) ensures that the partitions are hierarchical.

In the LP, where fractional values are allowed, $x_{i,j}^{(t)}$ is said to be the *LP-distance* between i, j at level t. If their LP-distance is small, then one should think of it as the LP suggesting that i, j should be in the same part of the output partition, while a large LP-distance suggests that they should not. Notice that for any given level t, the LP-distances satisfy the triangle inequality, by inequality (5).

We also note that the Correlation Clustering problem directly corresponds to the case where $\ell = \delta_1 = 1$.

3 FROM HIERARCHICAL CORRELATION CLUSTERING TO HIERARCHICAL CLUSTER AGREEMENT PROBLEM

Our main technical contribution is proving the following theorem.

THEOREM 3.1. *The Hierarchical Correlation Clustering problem can be solved in deterministic polynomial time within a constant approximation factor.*

In this section, we present a deterministic reduction from Hierarchical Correlation Clustering to Hierarchical Cluster Agreement that guarantees:

$$\text{HierCorrClust} \leq (\text{CorrClust} + 1)(\text{HierClustAgree} + 1) - 1. \qquad \text{(C) from Figure 1}$$

In Sections 4 and 5 we present a deterministic polynomial time constant factor approximation algorithm for Hierarchical Cluster Agreement; combined with a known deterministic polynomial time constant factor approximation algorithm for Correlation Clustering [52], it completes the proof of Theorem 3.1.

Assume that Correlation Clustering can be approximated within a factor α and that Hierarchical Cluster Agreement can be approximated within a factor β (Section 4). We prove Inequality (C), by providing an algorithm to approximate Hierarchical Correlation Clustering within a factor $(\alpha+1)(\beta+1)-1$.

Suppose we have a Hierarchical Correlation Clustering instance $S, \delta^{(1)}, \dots, \delta^{(\ell)}, E^{(1)}, \dots, E^{(\ell)}$. Our algorithm first solves the Correlation Clustering instance $S, E^{(t)}$ to acquire partition $Q^{(t)}$, for every level t. Then, we solve the Hierarchical Cluster Agreement instance $S, \delta^{(1)}, \dots, \delta^{(\ell)}$, $\mathcal{E}(Q^{(1)}), \dots, \mathcal{E}(Q^{(\ell)})$ and obtain hierarchical partitions $P^{(1)}, \dots, P^{(\ell)}$.

The proof that the hierarchical partitions $P^{(1)}, \dots, P^{(\ell)}$ are a good approximation to the Hierarchical Correlation Clustering instance follows from two observations. First, by definition, the cost of Hierarchical Correlation Clustering is related to certain symmetric differences. Since the cardinality of symmetric differences satisfy the triangle inequality, as seen by the connection to the Hamming distance, we can switch between the cost of Hierarchical Correlation Clustering and Hierarchical Cluster Agreement under the same output, with only an additive term related to $|E^{(t)} \triangle \mathcal{E}(Q^{(t)})|$ and not related to the output. Second, by definition of $Q^{(t)}$, the cardinality of the symmetric difference $|E^{(t)} \triangle \mathcal{E}(Q^{(t)})|$ is minimized within a factor α.

More formally, for this proof we need to define the following three concepts:

— For any $t \in [\ell]$, $OPT^{(t)}_{CorrClust}$ is an optimal solution to the Correlation Clustering instance at level t, that is, a partition minimizing

$$\left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{CorrClust}\right)\right|.$$

— $OPT_{HierCorrClust} = (OPT^{(1)}_{HierCorrClust}, \dots, OPT^{(\ell)}_{HierCorrClust})$ is an optimal solution to the Hierarchical Correlation Clustering instance, that is, a sequence of hierarchical partitions minimizing

$$\sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right|.$$

— $OPT_{HierClustAgree} = (OPT^{(1)}_{HierClustAgree}, \dots, OPT^{(\ell)}_{HierClustAgree})$ is an optimal solution to the Hierarchical Cluster Agreement instance, that is, a sequence of hierarchical partitions minimizing

$$\sum_{t=1}^{\ell} \delta^{(t)} \left|\mathcal{E}(Q^{(t)}) \triangle \mathcal{E}\left(OPT^{(t)}_{HierClustAgree}\right)\right|.$$

Notice, for any t, the difference between $OPT^{(t)}_{CorrClust}$ and $OPT^{(t)}_{HierCorrClust}$. The first one optimizes locally (per level), meaning that $|E^{(t)} \triangle \mathcal{E}(OPT^{(t)}_{CorrClust})| \le |E^{(t)} \triangle \mathcal{E}(OPT^{(t)}_{HierCorrClust})|$, and therefore

$$\sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{CorrClust}\right)\right| \le \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right|.$$

This does not contradict the definition of $OPT_{HierCorrClust}$, as $OPT^{(1)}_{CorrClust}, \dots, OPT^{(\ell)}_{CorrClust}$ is not a sequence of hierarchical partitions.

The cost of our solution is

$$\sum_{t=1}^{\ell} \delta^{(t)}|E^{(t)} \triangle \mathcal{E}(P^{(t)})| \leq \sum_{t=1}^{\ell} \delta^{(t)}|E^{(t)} \triangle \mathcal{E}(Q^{(t)})| + \sum_{t=1}^{\ell} \delta^{(t)}|\mathcal{E}(Q^{(t)}) \triangle \mathcal{E}(P^{(t)})|. \tag{8}$$

By definition of $P^{(1)}, \ldots, P^{(\ell)}$, they minimize the second term of inequality (8) within a factor β. Therefore, the second term can be rewritten as $\beta \sum_{t=1}^{\ell} \delta^{(t)}|\mathcal{E}(Q^{(t)}) \triangle \mathcal{E}(OPT^{(t)}_{HierClustAgree})|$, which, by optimality of $OPT_{HierClustAgree}$, is upper bounded by $\beta \sum_{t=1}^{\ell} \delta^{(t)}|\mathcal{E}(Q^{(t)}) \triangle \mathcal{E}(OPT^{(t)}_{HierCorrClust})|$.

Using the triangle inequality again, we further upper bound the second term by

$$\beta \sum_{t=1}^{\ell} \delta^{(t)}|\mathcal{E}(Q^{(t)}) \triangle E^{(t)}| + \beta \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right|.$$

Therefore, we can rewrite inequality (8) as

$$\sum_{t=1}^{\ell} \delta^{(t)}|E^{(t)} \triangle \mathcal{E}(P^{(t)})| \leq (\beta + 1) \sum_{t=1}^{\ell} \delta^{(t)}|E^{(t)} \triangle \mathcal{E}(Q^{(t)})| + \beta \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right|.$$

Since $Q^{(t)}$ is obtained by solving Correlation Clustering at level t within a factor α, we get

$$\sum_{t=1}^{\ell} \delta^{(t)}|E^{(t)} \triangle \mathcal{E}(Q^{(t)})| \leq \alpha \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{CorrClust}\right)\right|.$$

By optimality of $OPT^{(t)}_{CorrClust}$, for each $t \in [\ell]$, we have

$$\sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{CorrClust}\right)\right| \leq \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right|,$$

which proves that

$$\begin{aligned}\sum_{t=1}^{\ell} \delta^{(t)}|E^{(t)} \triangle \mathcal{E}(P^{(t)})| &\leq ((\beta + 1)\alpha + \beta) \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right| \\ &= ((\alpha + 1)(\beta + 1) - 1) \sum_{t=1}^{\ell} \delta^{(t)} \left|E^{(t)} \triangle \mathcal{E}\left(OPT^{(t)}_{HierCorrClust}\right)\right|.\end{aligned}$$

4 CONSTANT APPROXIMATION ALGORITHM FOR HIERARCHICAL CLUSTER AGREEMENT

In this section, we introduce our main algorithm, which consists of three parts: Solving the LP formulation of the problem, the LP-Cleaning subroutine, and the Derive-Hierarchy subroutine.

Informally, the LP-Cleaning subroutine uses the fractional solution of the LP relaxation of Hierarchical Cluster Agreement to decide which of our input-sets are important and which are not. The decision is not a binary one, because important sets are also cleaned, in the sense that bad parts of them may be removed. However, at least a 0.9 fraction of them is left intact, while unimportant sets are completely discarded.

The Derive-Hierarchy part then receives the cleaned input-sets by LP-Cleaning and applies a very simple combinatorial algorithm on them to compute the output.

We notice that the weights $\delta^{(*)}$ are only used for solving the LP. Moreover, the fractional LP-solution is only used by LP-Cleaning to guide this "nearly-binary" decision for each input-set. The rest of the algorithm is combinatorial and does not take the LP-solution into account.

4.1 LP Definitions for Hierarchical Cluster Agreement

The IP-formulation of Hierarchical Cluster Agreement is akin to the IP-formulation of Hierarchical Correlation Clustering. Namely, the constraints are exactly the same for both problems. The only difference is in the objective function where we replace the general edge-sets $E^{(1)}, \dots, E^{(\ell)}$ with the disjoint clique edges from $\mathcal{E}(Q^{(1)}), \dots, \mathcal{E}(Q^{(\ell)})$ and similarly for the LP-relaxation of Hierarchical Cluster Agreement. Here each component in $Q^{(t)}$ is called a *level-t input-cluster*.

To simplify our discussion, we use $x^{(*)}$ to denote a fractional solution to the LP-relaxation of Hierarchical Cluster Agreement, that is, a vector containing all $x^{(t)}_{i,j}, \{i,j\} \in \binom{S}{2}, t \in [\ell]$. One can think of $x^{(*)}$ as the optimal fractional solution, but in principle it can be any solution.

We use $x^{(t)}$, for some particular $t \in [\ell]$, to denote the vector containing all $x^{(t)}_{i,j}, \{i,j\} \in \binom{S}{2}$.

As previously, we use the term LP distances to refer to the entries of $x^{(*)}$ and notice that for any particular $t \in [l]$ the LP distances even satisfy the triangle inequality by the LP constraints.

Given $x^{(*)}$, we define $B^{(t)}_{<r}(i)$ to be the ball of species with LP-distance less than r from i at level t. More formally, $B^{(t)}_{<r}(i) = \{j \in S \mid x^{(t)}_{i,j} < r\}$. Similarly, for a subset S' of S we define the ball $B^{(t)}_{<r}(S') = \{j \in S \mid \exists i \in S' \text{ s.t. } x^{(t)}_{i,j} < r\}$.

We also define the LP cost of species i, j at level t as

$$cost^{(t)}_{i,j} = \begin{cases} \delta^{(t)} x^{(t)}_{i,j} \text{ if } \{i,j\} \in \mathcal{E}(Q^{(t)}) \\ \delta^{(t)}(1 - x^{(t)}_{i,j}) \text{ otherwise} \end{cases},$$

as well as the LP cost of species in a set $S' \subseteq S$ at level t as

$$cost^{(t)}_{S'} = \sum_{\substack{\{i,j\} \in \binom{S}{2} \\ i \in S' \text{ or } j \in S'}} cost^{(t)}_{i,j},$$

and in case S' only contains a single species i, we write $cost^{(t)}_i$ instead of $cost^{(t)}_{\{i\}}$.

Then the LP cost at level t is denoted as $cost^{(t)} = cost^{(t)}_S$.

Finally, the LP cost is simply $cost^{(*)} = \sum_{t=1}^{\ell} cost^{(t)}$.

4.2 Main Algorithm

The pseudocode for our main algorithm for Hierarchical Cluster Agreement is given in Algorithm 1.

ALGORITHM 1: Hierarchical Cluster AgreementAlgorithm

```
Input    A set $S$, a sequence $Q^{(*)} = (Q^{(1)}, \dots, Q^{(\ell)})$ of partitions of $S$, and weights
         $\delta^{(*)} = (\delta^{(1)}, \dots, \delta^{(\ell)})$
Returns  A sequence $P^{(*)} = (P^{(1)}, \dots, P^{(\ell)})$ of hierarchical partitions of $S$
1: $x^{(*)} \leftarrow$ Solve(LP-relaxation($S, Q^{(*)}, \delta^{(*)}$))
2: $L^{(*)} \leftarrow$ LP-Cleaning($S, Q^{(*)}, x^{(*)}$)
3: return Derive-Hierarchy($S, L^{(*)}$)
```

Our LP relaxation has size polynomial in S, ℓ, and the two subroutines also run in polynomial time, as we show later. Therefore, the whole algorithm runs in polynomial time.

4.3 LP Cleaning Algorithm

In Algorithm 2, we provide the pseudocode of the LP Cleaning step of our algorithm.

Intuitively, the aim of this algorithm is to clean the input sets so that (ideally) all species remaining in a set have small LP distances to each other and large LP distances to species not in the set.

ALGORITHM 2: LP-Cleaning

```
Input     A set $S$, a sequence $Q^{(*)} = (Q^{(1)}, \dots, Q^{(\ell)})$ of partitions of $S$,
          and a fractional solution $x^{(*)}$
Returns   A sequence $L^{(*)} = (L^{(1)}, \dots, L^{(\ell)})$ of families of disjoint subsets of $S$
1: for $t \leftarrow 1, \dots, \ell$ do
2:     $L^{(t)} \leftarrow \emptyset$
3:     for $C_I \in Q^{(t)}$ do
4:         $C_{LP} \leftarrow \left\{ i \in C_I \;\middle|\; \begin{array}{l} |B^{(t)}_{<0.1}(i) \cap C_I| > \frac{1}{2}|C_I|, \\ |B^{(t)}_{<0.6}(i) \setminus C_I| \leq 0.05|C_I| \end{array} \right\}$
5:         if $|C_{LP}| \geq 0.9|C_I|$ then
6:             $L^{(t)} \leftarrow L^{(t)} \cup \{C_{LP}\}$
7: return $L^{(*)} = (L^{(1)}, \dots, L^{(\ell)})$
```

Formally, Algorithm 2 takes a sequence $Q^{(*)} = (Q^{(1)}, \dots, Q^{(\ell)})$ of partitions of S and a fractional solution $x^{(*)}$ containing LP distances. It outputs a sequence of families of disjoint subsets of S, $L^{(*)} = (L^{(1)}, \dots, L^{(\ell)})$. Here each component of $L^{(t)}$ is called a *level-t LP-cluster.*

In the algorithm, for each input partition $Q^{(t)}$ we process every level-t input-cluster $C_I \in Q^{(t)}$ separately. For this, we remove all the species in C_I that do not have very small LP distance to at least half the species in C_I or that have small LP distance to many species not in C_I. More formally, we remove all the species in C_I with LP distance less than 0.1 to at most half the species in C_I or with LP distance less than 0.6 to more than $0.05|C_I|$ species not in C_I.

After the cleaning step, we discard C_I if smaller than a $9/10$ fraction of the species survive. Otherwise, we create an LP-cluster C_{LP} containing the species in C_I that survive. Next, we add the level-t LP-cluster C_{LP} to $L^{(t)}$.

Of several properties that we prove concerning the output of the LP-Cleaning, we briefly mention the following one: The output sequence $L^{(*)}$ is *hierarchy-friendly* in the sense that no two LP-clusters at the same level t can be intersected by the same LP-cluster at level $t' < t$. We formally prove this in Lemma 5.4.

The LP-Cleaning subroutine trivially runs in time polynomial in S, ℓ.

4.4 Derive-hierarchy Algorithm

In this section, we introduce Derive-Hierarchy (Algorithm 3). It takes as input a hierarchy-friendly sequence $L^{(*)} = (L^{(1)}, \dots, L^{(\ell)})$ of families of disjoint subsets of S and outputs a sequence $P^{(*)} = (P^{(1)}, \dots, P^{(\ell)})$ of hierarchical partitions of S. The execution of the algorithm can be seen, via a graphical example, in Figure 2.

The algorithm works bottom-up while performing ℓ iterations for $t = 1, \dots, \ell$. In the process, it incrementally builds a forest $\mathcal{F}$. Throughout the algorithm, each non leaf node u in $\mathcal{F}$ can be identified by an LP-cluster in $L^{(*)}$. Moreover, for each node u the algorithm maintains two sets $C(u) \subseteq S$, called the core-cluster of u, and $C^+(u) \subseteq S$, called the extended-cluster of u.

The algorithm starts by initializing $\mathcal{F}$ with $|S|$ trees where each tree contains a single node u_i identified by a species $i \in S$. Also, it initializes both sets $C(u_i), C^+(u_i)$ with $\{i\}$. Next, in iteration t the algorithm processes the LP-clusters in $L^{(t)}$ and at the end of the iteration, the $C^+()$ sets

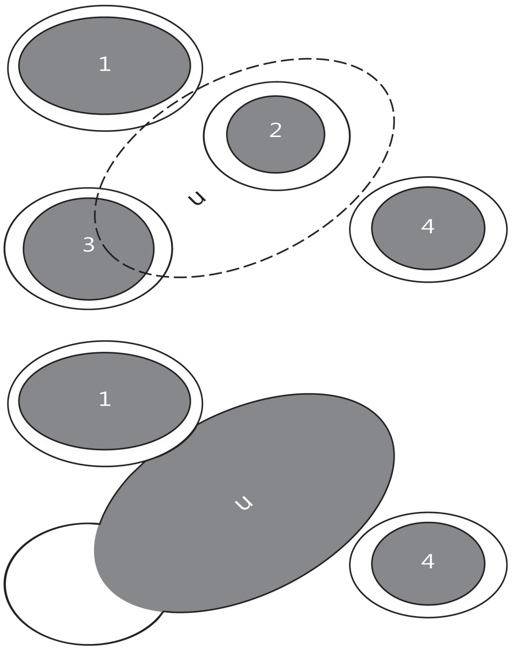

Fig. 2. Example of Derive-Hierarchy (Algorithm 3). Nodes $1, 2, 3, 4$ (left) are the roots of the forest before inserting the new LP-cluster $L(u)$ (dashed line). Each node is described by its extended-cluster, with the shaded part being the core-cluster. Core-cluster of nodes 2 and 3 intersect $L(u)$; thus they become children of u and the extended-cluster of u covers the extended-clusters of $2, 3$ (right). Notice that the core-cluster of u is reduced due to node 1.

associated with the root nodes in $\mathcal{F}$ define the partition $P^{(t)}$. Precisely here, the $C^+()$ set of a root node contains all the species descending from the respective root and will thus be the sets in the final partitions.

In the tth iteration, for each cluster $C_{LP} \in L^{(t)}$ the algorithm adds a root node u to forest $\mathcal{F}$ while initializing the set $C(u)$ with C_{LP}. Next for the root node u the algorithm decides on its children by processing the pre-existing roots in the following way. For consistency, first it detects all the pre-existing root nodes v such that $C(u)$ does not intersect $C(v)$. Then it removes from $C(u)$ all the species that are descending from v, i.e., sets $C(u) \leftarrow C(u) \setminus C^+(v)$. Last, it sets u as a parent of all other pre-existing root nodes v such that $C(u)$ intersects $C(v)$. Also, accordingly, it modifies the set of leaf nodes of the subtree rooted at u by setting $C^+(u) \leftarrow C^+(u) \cup C^+(v)$. Notice here that some of the root-nodes may correspond to sets from levels lower than t, in case no parent was assigned to them.

At the end of iteration t, the algorithm completes processing all the LP-clusters in $L^{(1)}, \ldots, L^{(t)}$ and constructs partitions $P^{(1)}, \ldots, P^{(t)}$. At the end of the ℓ iterations it outputs the ℓ partitions $P^{(*)} = (P^{(1)}, \ldots, P^{(\ell)})$.

The Derive-Hierarchy subroutine trivially runs in time polynomial in S, ℓ.

5 ANALYSIS OF HIERARCHICAL CLUSTER AGREEMENT ALGORITHM

In this section, we proceed with our analysis. We first lay out some terminology and then provide some results related to the LP Cleaning, then some structural results, and finally prove that our algorithm is a constant factor approximation for Hierarchical Cluster Agreement.

ALGORITHM 3: Derive-Hierarchy

```
Input     A set $S$, and a hierarchy-friendly sequence $L^{(*)} = (L^{(1)}, \dots, L^{(\ell)})$
          of families of disjoint subsets of $S$
Returns   A sequence $P^{(*)} = (P^{(1)}, \dots, P^{(\ell)})$ of hierarchical partitions of $S$
1:  Construct an empty forest $\mathcal{F}$
2:  for $i \in S$ do
3:      Create a singleton tree $T$ with a node $u_i$ and add it to $\mathcal{F}$
4:      Set $C(u_i) \leftarrow \{i\}, C^{+}(u_i) \leftarrow \{i\}$
5:  for $t \leftarrow 1, \dots, l$ do
6:      for $C_{LP} \in L^{(t)}$ do
7:          Create a node $u$ and set $C(u) \leftarrow C_{LP}$
8:          for all roots $v \in \mathcal{F}$ s.t. $C(v) \cap C(u) = \emptyset$ do
9:              $C(u) \leftarrow C(u) \setminus C^{+}(v)$
10:         $C^{+}(u) \leftarrow C(u)$
11:         for all roots $v \in \mathcal{F}$ s.t. $C(v) \cap C(u) \neq \emptyset$ do
12:             $C^{+}(u) \leftarrow C^{+}(u) \cup C^{+}(v)$
13:             Make $v$ a child of $u$ in $\mathcal{F}$
14:     Set $P^{(t)}$ to contain the extended-clusters $C^{+}(v)$ of all roots $v \in \mathcal{F}$
15: return $P^{(*)} = (P^{(1)}, \dots, P^{(\ell)})$
```

5.1 Terminology

Notice that throughout the execution of the algorithm, $\mathcal{F}$ is an incrementally updated graph (that is, no deletions occur). In fact, it is always a forest, as we start with $|S|$ isolated nodes and only introduce new nodes as parents of roots of some of the existing trees. Moreover, this process implies that the subtree rooted at any specific node is never modified.

From now on, we use $\mathcal{F}$ to refer to the final instance of the incrementally updated forest. We use $\mathcal{F}(u)$ to refer to the state of this incrementally updated forest after introducing u; therefore, $\mathcal{F}(u) \setminus \{u\}$ denotes the state of the forest exactly before introducing node u. We naturally identify the leaves of $\mathcal{F}$ with the species of S.

For any node u in the forest $\mathcal{F}$, the Derive-Hierarchy algorithm defines $C(u)$, which we call the core-cluster of u, and $C^{+}(u)$, which we call the extended-cluster of u. Furthermore, notice that each core-cluster $C(u)$ is a subset of some LP-cluster C_{LP} (Lines 7–9 of Algorithm 3); we call this the LP-cluster of u and denote it by $L(u)$. Moreover, each LP-cluster $L(u)$ is a subset of an input-cluster C_I (Line 3 of Algorithm 2); we call this the input-cluster of u and denote it by $I(u)$. These concepts are well defined for any new node u and never change throughout the algorithm. We remind the reader that LP-Cleaning discards some of the input-clusters, in the sense that they have no corresponding LP-cluster, and therefore they do not match $I(u)$, for any node u.

Directly from the algorithm we get that

$$C(u) \subseteq L(u) \subseteq I(u)$$
$$C(u) \subseteq C^{+}(u).$$

To help with our discussion, we also define the following variables related to the Derive-Hierarchy algorithm (Algorithm 3):

$$\Delta^{-}(u) = L(u) \quad \setminus C(u)$$
$$\Delta^{+}(u) = C^{+}(u) \setminus C(u).$$

For a node $u \in \mathcal{F}$, we define its level $t(u)$ to be the value of iteration t in Algorithm 3 when internal node u was introduced, and 0 when u is a leaf node.

5.2 LP-Cleaning Results (Algorithm 2)

We start with some observations that are heavily used in proving structural results regarding the core and the extended-clusters. These are in turn used for lower-bounding the LP cost.

The most important reason we are using the LP-Cleaning subroutine is so that any two species belonging in the same LP-Cluster at level t have small LP-distance.

Lemma 5.1. *Given a node $u \in \mathcal{F}$ and a species i in u's LP-cluster $L(u)$, it holds that the LP-distance from i to any other species in $L(u)$ is less than 0.2 for all levels $t \geq t(u)$, that is, $B^{(t)}_{<0.2}(i) \supseteq L(u)$.*

Proof. It suffices to prove that $x^{(t)}_{i,j} < 0.2$ for all $j \in L(u)$ only for level $t = t(u)$, as the LP constraints enforce $x^{(t+1)}_{i,j} \leq x^{(t)}_{i,j}$.

Since both $B^{(t)}_{<0.1}(i) \cap I(u)$ and $B^{(t)}_{<0.1}(j) \cap I(u)$ have size more than $|I(u)|/2$ (Line 4 of Algorithm 2), there exists a node $k \in I(u)$ for which both $x^{(t)}_{i,k}$ and $x^{(t)}_{j,k}$ are less than 0.1. Since the LP-distances in $x^{(t)}$ satisfy the triangle inequality, it follows that $x^{(t)}_{i,j} < 0.2$ (enforced by the LP constraints). □

For the analysis, it is convenient that our relations involve the LP-clusters instead of the input-clusters. Therefore, we rephrase Line 4 of Algorithm 2 in terms of LP-clusters, effectively proving that few species outside of an LP-cluster $L(u)$ have small LP-distances to $L(u)$.

Lemma 5.2. *For any node $u \in \mathcal{F}$ it holds that $|B^{(t(u))}_{<0.4}(L(u))| \leq (1 + \frac{1}{6})|L(u)|$. In particular, $|B^{(t(u))}_{<0.4}(L(u)) \setminus L(u)| \leq \frac{1}{6}|L(u)|$.*

Proof. Let $t = t(u)$. We claim that species close to some species in $L(u)$ are still *reasonably* close to all species in $L(u)$. Formally, we claim that for any $i \in L(u)$

$$B^{(t)}_{<0.4}(L(u)) \subseteq B^{(t)}_{<0.6}(i).$$

Let $j \in B^{(t)}_{<0.4}(L(u))$. We bound the LP-distance between i, j by finding an intermediate i' that is close to both and applying the triangle inequality forced by the LP constraints. By definition of j, there exists a species $i' \in L(u)$ with LP-distance less than 0.4 from j. By Lemma 5.1, the LP-distance between i and i' is less than 0.2, and thus by triangle inequality $x^{(t)}_{i,j} < 0.6$.

Line 4 of Algorithm 2 gives that

$$|B^{(t)}_{<0.6}(i) \setminus I(u)| \leq 0.05|I(u)| \implies |B^{(t)}_{<0.6}(i)| \leq 0.05|I(u)| + |I(u)|.$$

Combining these two relations, and by $|L(u)| \geq 0.9|I(u)|$ (Line 5 of Algorithm 2),

$$|B^{(t)}_{<0.4}(L(u))| \leq |B^{(t)}_{<0.6}(i)| \leq \frac{(1+0.05)}{0.9}|L(u)| = \left(1 + \frac{1}{6}\right)|L(u)|.$$ □

The following lemma is just a convenient application of the triangle inequality of our LP, that is heavily used in subsequent proofs. Informally, it states that, under certain mild conditions, the LP-distance is small not only if i, j belong in the same LP-cluster (or core-cluster), but even if they happen to be in different clusters that are both intersected by the same third cluster.

Lemma 5.3. *Let $u, v, w \in \mathcal{F}$ be three arbitrary nodes. Assume that the LP-cluster of v intersects the LP-clusters of u and w and $t_{max} = \max\{t(u), t(v), t(w)\}$. Then for any $i, j \in \{L(u) \cup L(v) \cup L(w)\}$ their LP-distance at level t_{max} is less than 0.6, and $B^{(t_{max})}_{<0.4}(L(u)) \supseteq L(u) \cup L(v) \cup L(w)$.*

Proof. If both i, j are in $L(u)$, $L(v)$, or $L(w)$, then the claim follows trivially from Lemma 5.1. Otherwise, we use triangle inequality twice, with species in the intersections of the clusters as intermediates. More formally, let $k \in L(u) \cap L(v)$, $k' \in L(v) \cap L(w)$. Lemma 5.1 implies three things:

(1) $x_{i,k}^{(t_{max})} < 0.2$, for any $i \in L(u) \cup L(v)$
(2) $x_{k,k'}^{(t_{max})} < 0.2$, as both $k, k' \in L(v)$
(2) $x_{k',j}^{(t_{max})} < 0.2$, for any node $j \in L(v) \cup L(w)$.

Since the LP-distances $x^{(t_{max})}$ respect the triangle inequality, it holds that $x_{i,j}^{(t_{max})} < 0.6$. The claim about the ball of $L(u)$ follows by taking the distance from k to j. □

We are now ready to prove the hierarchy-friendly property of the output of LP-Cleaning, as we informally claimed when introducing the algorithm. We claim that two LP-clusters of the same level cannot be intersected by the same lower level LP-cluster.

Lemma 5.4. *Given two nodes $v, w \in \mathcal{F}$ on the same level, there is no lower level node u such that $L(u)$ intersects both $L(v)$ and $L(w)$.*
In particular, there is also no $C(u)$ intersecting both $L(v)$ and $L(w)$.

Proof. The intuition is that $L(v), L(w)$ are close and thus Algorithm 2 would discard at least one of them.
Without loss of generality, let $|L(v)| \geq |L(w)|$. $L(v), L(w)$ are disjoint, as they are subsets of different parts of the partition $Q^{(t(v))}$, by Algorithm 2.
By Lemma 5.3 $|B_{<0.4}^{(t(w))}(L(w))| \geq |L(v)| + |L(w)| \geq 2|L(w)|$, which contradicts Lemma 5.2. □

We finally present a simple lower bound on the LP cost. Recall that

$$cost_{i,j}^{(t)} = \begin{cases} \delta^{(t)} x_{i,j}^{(t)} & \text{if } \{i,j\} \in \mathcal{E}(Q^{(t)}) \\ \delta^{(t)}(1 - x_{i,j}^{(t)}) & \text{otherwise} \end{cases}$$

and

$$cost_{S'}^{(t)} = \sum_{\substack{\{i,j\} \in \binom{S}{2} \\ i \in S' \text{ or } j \in S'}} cost_{i,j}^{(t)}.$$

Lemma 5.5. *Let $C_I \in Q^{(t)}$ be an input-cluster at level t, and C_{LP} be the respective LP-cluster from Algorithm 2. Fix a species $i \in C_I \setminus C_{LP}$. Then the fractional LP cost $cost_i^{(t)} = \Omega(\delta^{(t)}|C_I|)$.*

Proof. There are two reasons for i to be in $C_I \setminus C_{LP}$, by Line 4 of Algorithm 2. Either half the species in C_I are at distance at least 0.1 from i or more than $0.05|C_I|$ species not in C_I are at distance at most 0.6 from i.
In the first case, $cost_i^{(t)} \geq 0.1 \cdot (\frac{1}{2}|C_I|)\delta^{(t)}$, and in the second case, $cost_i^{(t)} \geq (1 - 0.6) \cdot 0.05|C_I|$ $\delta^{(t)}$. □

5.3 Derive-Hierarchy results (Algorithm 3)

In this section, we present several structural results related to our algorithm.
We start with pointing out that our algorithm ends up with the same output, no matter the order in which we process LP-clusters of the same level. This is due to the input sequence $L^{(*)}$ being hierarchy-friendly.

Lemma 5.6. *The output of Algorithm 3 is the same, irrespective of the order in which LP-clusters of the same level are processed in Line 6.*

Proof. For each level, fix any ordering in which LP-clusters of the same level are processed and run the algorithm. For any $t \in [\ell]$, let $\mathcal{F}_{t-1}$ be the state of the forest just before processing the first node of level t. We show that for any level-t LP-cluster C_{LP} with corresponding node u (that is $t(u) = t$ and $L(u) = C_{LP}$), no matter when it was actually processed due to the ordering we fixed, the effect is the same as if it was the first level-t LP-cluster processed. More formally, let $N(u)$ be the set of children of u, and $C_{t-1}(u)$, $C^+_{t-1}(u)$, and $N_{t-1}(u)$ be the core-cluster, the extended-cluster, and the set of children of u in the case where C_{LP} was the first LP-cluster of level-t to be processed. Then $C(u) = C_{t-1}(u)$, $C^+(u) = C^+_{t-1}(u)$, and $N(u) = N_{t-1}(u)$.

The main idea is that if a root $v \in \mathcal{F}_{t-1}$ has a core-cluster intersecting $L(u)$, then it is still a root just before inserting u; else v would have another parent w of level t, meaning $C(v) \subseteq L(v)$ would also intersect $C(w) \subseteq L(w)$ (Line 11), which contradicts that $L^{(*)}$ is hierarchy-friendly.

For u's children, we first show that $N(u) \subseteq N_{t-1}(u)$. Suppose this was not true, then there would exist a level-t node $v \in N(u) \setminus N_{t-1}(u)$. That would imply that u's and v's core clusters intersect (Line 11). But core-clusters are always subsets of their corresponding LP-clusters, and LP-clusters of the same level are disjoint.

Before proving $N_{t-1}(u) \subseteq N(u)$, we need to show that $C(u) = C_{t-1}(u)$. We show it by proving that $L(u) \setminus C(u) = L(u) \setminus C_{t-1}(u)$. If a species i is in $L(u) \setminus C(u)$, then it is in the extended-cluster of some node v processed before u such that their core-clusters do not intersect (Line 8). If $t(v) = t$, then i is either in $C(v)$ (contradiction as it would then not be in $L(u)$) or in the extended-cluster of one of its children w, which we proved are of lower-level. Thus, w was a root in $\mathcal{F}_{t-1}$. Again, by $L^{(*)}$ being hierarchy-friendly, $C(w) \subseteq L(w)$ does not intersect $L(u)$, meaning it does not intersect $C_{t-1}(u) \subseteq L(u)$, and so i would also be in $L(u) \setminus C_{t-1}(u)$ (Line 8). If $t(v) < t$, then v itself was a root in $\mathcal{F}_{t-1}$. The same argument in reverse order is used to prove that if i is in $L(u) \setminus C_{t-1}(u)$, then it is in $L(u) \setminus C(u)$.

We now see that $N_{t-1}(u) \subseteq N(u)$; that is, because if $v \in N_{t-1}(u)$, then v is a root in $\mathcal{F}_{t-1}$ with a core-cluster intersecting $L(u)$, and by $L^{(*)}$ being hierarchy-friendly it is also a root in $\mathcal{F}(u) \setminus \{u\}$. As $C(v)$ intersects $C_{t-1}(u) = C(u)$, we get $v \in N(u)$ (Line 11).

Finally, a species i in $C^+_{t-1}(u) \setminus C_{t-1}(u)$ is part of the extended cluster of a node in $N_{t-1}(u)$; this child is still a root in $\mathcal{F}(u) \setminus \{u\}$ by $L^{(*)}$ being hierarchy-friendly; therefore, $i \in C^+(u) \setminus C(u)$. The other way around, a species i in $C^+(u) \setminus C(u)$ is part of the extended cluster of a node in $N(u)$; this child is a root in $\mathcal{F}_{t-1}$ as u has no children of level t; therefore, $i \in C^+(u) \setminus C_{t-1}(u)$. □

Next, we prove two claims that we have already mentioned informally while introducing Algorithm 3 (Derive-Hierarchy). First we claim that the incrementally built graph is always a forest and also for any node u in F, its extended-cluster contains exactly the species that are identified with leaves descending from u in $\mathcal{F}$. We call these species the *descending species* of u. We note that the results of Section 5.2 do not require these properties.

Lemma 5.7. *For any $u \in \mathcal{F}$, $\mathcal{F}(u)$ is a forest of rooted trees with $|S|$ leaves identified with the species of S. Moreover, for each $u \in \mathcal{F}$, $C^+(u)$ is the set of u's descending species.*

Proof. We prove this inductively based on the order in which the nodes are added to $\mathcal{F}$. The base case for both the claims follows by the initialization of the forest with $|S|$ leaves identified with the species of S.

When we insert a node, it becomes the parent of some of the existing roots, and therefore the forest structure is preserved.

Next let u be some node in $\mathcal{F}$, and let $v_1, \ldots, v_k$ be the children of u. Then, by construction, all these children nodes are added to $\mathcal{F}$ before u, and thus by an inductive argument for each v_m the set of descending species of v_m is exactly the set $C^+(v_m)$. Now we need to prove the same for node u. Note that the set of descending species of u is precisely the set $\bigcup_{m\in[k]} C^+(v_m)$. Moreover, by construction, $C^+(u) = C(u) \cup (\bigcup_{m\in[k]} C^+(v_m))$. Hence, to prove the claim, we need to show that $C(u) \subseteq \bigcup_{m\in[k]} C^+(v_m)$. For the sake of contradiction, let $w \in C(u)$ be a species such that $w \notin \bigcup_{m\in[k]} C^+(v_m)$. As $\mathcal{F}(u) \setminus \{u\}$ is a forest, there exists a unique node r that is the root node of the tree of $\mathcal{F}(u) \setminus \{u\}$ that contains w. Hence, again by induction argument $w \in C^+(r)$. By our assumption, as r is not a child of u, $C(u) \cap C(r) = \emptyset$. Thus, Algorithm 3 (Line 8) sets $C(u) \leftarrow C(u) \setminus C^+(r)$, and hence $w \notin C(u)$, which is a contradiction. □

This simple lemma alone is enough to prove the following corollaries:

Corollary 5.8. *For any $u \in \mathcal{F}$, the extended-clusters of the root nodes in $\mathcal{F}(u)$ form a partition of S.*

Proof. As $\mathcal{F}(u)$ is a forest, each species is a descendant of exactly one such root and thus belongs in exactly one such extended cluster. □

Corollary 5.9. *The output of our algorithm is a sequence of hierarchical partitions of S.*

Proof. By Corollary 5.8, the output of the algorithm is a sequence of partitions of S. To see that the output partitions are hierarchical, notice that if two species are in the same rooted tree at some point in the algorithm, then they are never separated as we only add nodes in the forest. □

Corollary 5.10. *For any node $u \in \mathcal{F}$, the species removed from its LP-cluster and the species inserted in its core cluster are disjoint, $\Delta^-(u) \cap \Delta^+(u) = \emptyset$.*

Proof. For the sake of contradiction, let $i \in \Delta^-(u) \cap \Delta^+(u)$. Since the extended clusters of root nodes in $\mathcal{F}(u) \setminus \{u\}$ form a partition, let v be the unique such root for which $i \in C^+(v)$. Now as $i \in \Delta^-(u)$, $C(v) \cap C(u) = \emptyset$ (Line 8 of Algorithm 3). But, again, $i \in \Delta^+(u)$ implies $C(v) \cap C(u) \neq \emptyset$ (Line 11 of Algorithm 3), and both of these can never be satisfied together. □

Corollary 5.11. *If two nodes $u, v \in \mathcal{F}$ do not have an ancestry-relationship, then their extended clusters do not intersect.*

Proof. If their extended clusters intersected, then they would have a descending species in common, which implies an ancestry-relationship. □

We also need the following result.

Lemma 5.12. *For any node $u \in \mathcal{F}$, its extended-cluster is equal to the union of the core clusters of all descendant nodes v of u.*

Proof. We prove this inductively. As a base-case, the claim trivially holds for the $|S|$ initial leaves. For an internal node u, let $v_1, \dots, v_k$ be the children of u and let $D(u)$ be the descendant nodes of u. Then $D(u) = \cup_{m\in[k]} D(v_m)$. Also, by induction, for each v_m, $C^+(v_m) = \cup_{w\in D(v_m)} C(w)$. Now as $C^+(u) = C(u) \cup (\cup_{m\in[k]} C^+(v_m))$ we have $C^+(u) = C(u) \cup (\cup_{w\in D(u)} C(w))$, which proves our claim. □

5.4 Managing Removals and Extensions

Using the developed toolkit of structural results, we are ready to show that for any node $u \in \mathcal{F}$, all three of the LP-cluster $L(u)$, the core-cluster $C(u)$, and the extended-cluster $C^+(u)$ are similar; more than that, we show lower bounds of the LP cost related to $\Delta^-(u) = L(u) \setminus C(u)$ and $\Delta^+(u) = C^+(u) \setminus C(u)$.

In particular, we claim that the following inequality holds for every $u \in \mathcal{F}$,

$$|\Delta^+(u)| \leq 0.3|C(u)|. \tag{9}$$

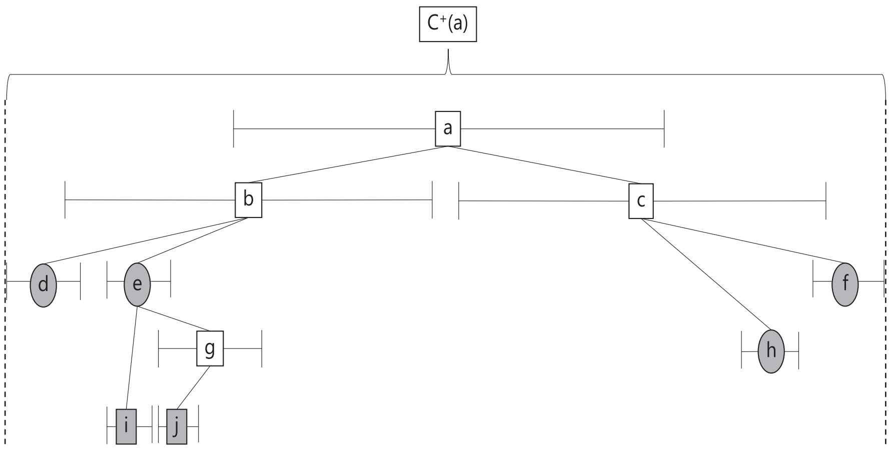

Fig. 3. Part of the forest $\mathcal{F}$(a). Intervals around nodes denote core-clusters (two core-clusters intersect if a vertical line intersects both); colored nodes $\{d, e, f, h, i, j\}$ have core-clusters not intersecting the core-cluster of a. In particular, the circle-shaped colored nodes are a's top-non-intersecting descendants (denoted by $J = \{d, e, f, h\}$). Their proper descendants define $J^+ = \{g, i, j\}$. $R = \{b, c\}$ contains all other proper descendants of a.

We prove this claim inductively, based on the order in which nodes are added in $\mathcal{F}$. As a base case, we initially create a node u_i for each species $i \in S$ with $C(u_i) = C^+(u_i) = \{i\}$, meaning that $\Delta^+(u_i) = \emptyset$.

For any other node u, we argue about the size of its extended-cluster $C^+(u)$ in relation with the core-clusters of its descendants, as suggested by Lemma 5.12. We now partition the descendants of u in three parts and argue about each one of them.

For a node u, we define its top-non-intersecting descendants J as the set of highest level descendant nodes in $\mathcal{F}$ whose core-clusters do not intersect $C(u)$ (the reader is encouraged to consult Figure 3 before proceeding). More formally, using $v \prec_{\mathcal{F}} u$ to denote that v is a descendant of u in forest $\mathcal{F}$, we have

$$J = \left\{ v \in \mathcal{F} \;\middle|\; \begin{array}{l} v \prec_{\mathcal{F}} u \\ C(u) \cap C(v) = \emptyset \\ C(u) \cap C(w) \neq \emptyset, \forall w \text{ s.t. } v \prec_{\mathcal{F}} w \prec_{\mathcal{F}} u \end{array} \right\}.$$

Notice that, by definition, if two nodes v, w belong in u's top-non-intersecting descendants, then none is an ancestor of the other. Therefore u's top-non-intersecting descendants J naturally partitions the proper descendants of u in three parts: J itself, the set J^+ of proper descendants of nodes in J, and R containing the rest of the proper descendants of u (i.e., the proper descendants of u that are not descendants of any node in J). We also define sets of species related to these sets,

$$\begin{aligned} S_J &= \bigcup_{v \in J} C(v) && (10) \\ S_{J^+} &= \bigcup_{v \in J^+} C(v) \quad \setminus (C(u) \cup S_J) \\ S_R &= \bigcup_{v \in R} C(v) \quad \setminus (C(u) \cup S_J \cup S_{J^+}). \end{aligned}$$

The apparent asymmetry of not excluding $C(u)$ from S_J follows from the definition of u's top-non-intersecting descendants J; the core-clusters of nodes in J are disjoint from $C(u)$, meaning that S_J would be the same even if we excluded species in $C(u)$. Note that this is not the case for the core-clusters in J^+, as proper descendants of nodes in J might still intersect $C(u)$, as in Figure 3.

Notice that by Lemma 5.12 we have $C^+(u) = C(u) \cup (S_J \cup S_{J^+} \cup S_R)$, and thus

$$\Delta^+(u) = S_J \cup S_{J^+} \cup S_R. \tag{11}$$

If $v \in J \cup J^+ \cup R$, and its core-cluster does not intersect the core-cluster of u, then by definition of J we have that v is in descendants (not necessarily proper) of J. Therefore, $v \in J \cup J^+$, meaning that nodes in R have core-clusters that intersect $C(u)$. Furthermore, by definition, each node v in u's top-non-intersecting descendants has a parent whose core-cluster intersects $C(u)$. Therefore, for any species $i \in C(u)$ and any species $j \in S_J \cup S_R$, by Lemma 5.3 we have that their LP-distance is small, that is,

$$x_{i,j}^{(t(u))} < 0.6, \forall i \in C(u), j \in S_J \cup S_R. \tag{12}$$

Species in $S_J \cup S_R$ are not in $C(u)$, and so by Corollary 5.10 they are not in $L(u)$, as they belong in $\Delta^+(u)$. Then Lemma 5.2 gives

$$|S_J \cup S_R| < \frac{1}{6}|L(u)|. \tag{13}$$

We are left to argue about species in J^+, that is, in core-clusters of the descendants of u's top-non-intersecting descendants. By Lemma 5.12, these species all belong in the extended-clusters of u's top-non-intersecting descendants, $\bigcup_{v \in J} C^+(v) \supseteq S_J \bigcup S_{J^+}$. By Corollary 5.11, these extended-clusters are disjoint, and thus $S_{J^+} \subseteq \bigcup_{v \in J} \Delta^+(v)$. By the inductive hypothesis (9), we get

$$|S_{J^+}| \le 0.3|S_J|. \tag{14}$$

Therefore, by Inequality (13) we get that

$$|S_{J^+}| < \frac{1}{6} \cdot 0.3|L(u)| = \frac{1}{6} \cdot 0.3|C(u) \cup \Delta^-(u)|. \tag{15}$$

By Equations (13) and (15), we bound the size of $\Delta^+(u)$,

$$|\Delta^+(u)| < 1.3 \cdot \frac{1}{6}|L(u)|. \tag{16}$$

We are only left with bounding $|L(u)|$. For this, we prove that

$$|\Delta^-(u)| < 0.1|C(u)|, \tag{17}$$

which, combined with Equation (16), proves our initial claim, as we have

$$|\Delta^+(u)| < 1.3 \cdot \frac{1}{6}|L(u)| < 1.3 \cdot \frac{1}{6}|\Delta^-(u) \cup C(u)| \le 1.3 \cdot \frac{1}{6} \cdot 1.1|C(u)| < 0.3|C(u)|. \tag{18}$$

Before proving Equation (17), we make some definitions (see Figure 4). Roughly speaking, we want to identify an appropriate set K of nodes such that the union of their extended-clusters both contains $\Delta^-(u)$ and its cardinality is reasonably boundable. In fact, the nodes in K are descendants of roots of $\mathcal{F}(u) \setminus \{u\}$ that satisfy the condition in Line 8 of Algorithm 3 (i.e., nodes v such that $C(v) \cap C(u) = \emptyset$, and $C^+(v) \cap L(u) \neq \emptyset$).

We now give a formal constructive definition of the set K. Let M be the set containing all the non-descendants of u at level at most $t(u)$ whose core-clusters intersect $L(u)$. We define K' to be the set of parents of the nodes in M. Finally, K is obtained from K' by removing the nodes who

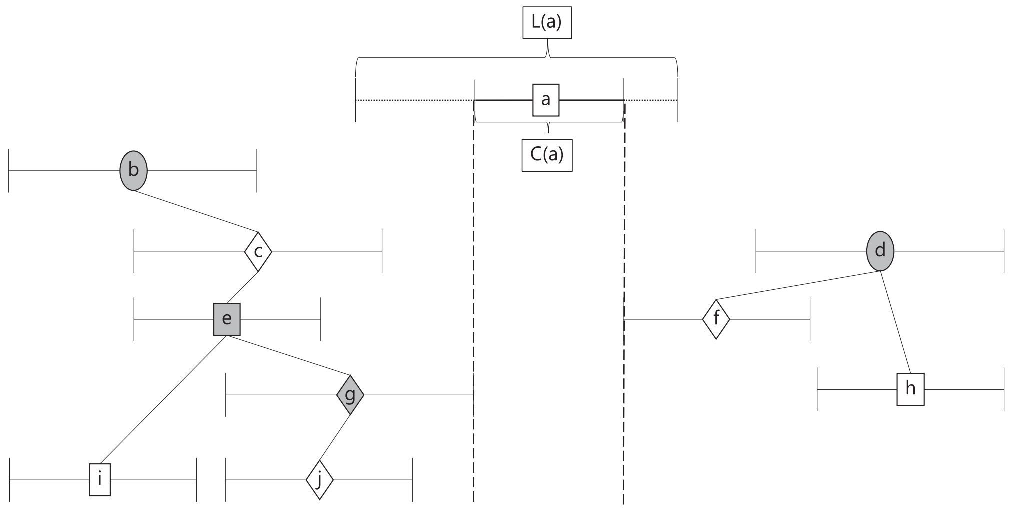

Fig. 4. Part of the forest $\mathcal{F}$(a). Intervals around nodes denote core-clusters (two core-clusters intersect if a vertical line intersects both); for node a we also denote its LP-cluster by horizontal dotted lines. All other depicted nodes are not descendants of a. The diamond-shaped nodes $\{c, f, g, j\}$ are contained in M, colored nodes $\{b, d, e, g\}$ are contained in K', and circle-shaped nodes $\{b, d\}$ are contained in $K \subseteq K'$.

have a proper ancestor in K'. Notice by Corollary 5.11 that their extended-clusters are disjoint. We also define sets of species associated with K as follows:

$$S_K = \bigcup_{v \in K} C(v), \tag{19}$$
$$S_{K^+} = \bigcup_{v \in K} C^+(v).$$

Note $\Delta^-(u) \subseteq S_{K^+}$. Next, we claim for each node $v \in K$, $C(v) \cap L(u) = \emptyset$. Now if we can prove this claim, then it implies $S_K \cap L(u) = \emptyset$, and thus we can write $\Delta^-(u) \subseteq S_{K^+} \setminus S_K$. Next, we prove the claim. Notice that for any node $v \in M$, v is not a descendant of u but $C(v) \cap L(u) \neq \emptyset$; thus there always exists a node $w \in K$ such that w is an ancestor of v and $C(w) \cap L(u) = \emptyset$.

Now, for the sake of contradiction, assume there exists a node $w \in K$ such that $C(w) \cap L(u) \neq \emptyset$. But then $w \in M$, and following the previous argument there exists a node $w' \in K$ such that w' is an ancestor of w and $C(w') \cap L(u) = \emptyset$. This is a contradiction, as by construction both w and w' cannot be present in K.

Furthermore, notice that no node $w \in K$ is at level $t(w) = t(u)$, as that would imply a child $w' \in M$ of w, but $C(w')$ intersects $C(w)$ (and therefore $L(w)$) as w' is a child of w, and $C(w')$ intersects $L(u)$, since $w \in M$. By Lemma 5.4, this is a contradiction.

We conclude that K contains nodes at level at most $t(u)-1$, which allows us to apply the inductive hypothesis $|\Delta^+(w)| \leq 0.3|C(w)|$ for nodes $w \in K$. Thus, from $\Delta^-(u) \subseteq S_{K^+} \setminus S_K$, we get

$$|\Delta^-(u)| \leq 0.3|S_K|. \tag{20}$$

Furthermore, all nodes in K have a child whose core-cluster intersects $C(u)$, and so by Lemma 5.3 the LP-distance between a species $i \in L(u)$ and a species $j \in S_K$ is small, $x_{i,j}^{t(u)} < 0.6$. By Lemma 5.2, we get $|S_K| < \frac{1}{6}|L(u)|$, which gives us $|\Delta^-(u)| \leq \frac{1}{6} \cdot 0.3|L(u)|$.

By the definition of $\Delta^-(u) = L(u) \setminus C(u)$, we get $|C(u)| \geq (1 - \frac{1}{6} \cdot 0.3)|L(u)|$, by which

$$|\Delta^-(u)| \leq \frac{\frac{1}{6} \cdot 0.3}{1 - \frac{1}{6} \cdot 0.3}|C(u)|, \tag{21}$$

which concludes the proof of claim (17) and, as previously argued, the proof of claim (9).

As a by-product of this analysis, we can also give some lower bounds on the LP cost.

Lemma 5.13. *Given a node $u \in \mathcal{F}$, $cost_{I(u)}^{(t(u))} = \Omega(\delta^{(t(u))}|L(u)||\Delta^-(u)|)$.*

Proof. Fix a $j \in S_K$, as defined in Equation (19). We have shown that for all $i \in L(u)$, the LP-distance with j is small, $x_{i,j}^{(t(u))} < 0.6$. If $j \in S_K$ is not in the input-cluster $I(u)$, then $cost_{i,j}^{(t(u))} = \delta^{(t(u))}(1 - x_{i,j}^{(t(u))}) > \delta^{(t(u))}(1 - 0.6)$ for each $\{i, j\}$ pair with $i \in L(u)$. Else, it was removed from the input-cluster in the LP-Cleaning step, $cost_j^{(t(u))} = \Omega(\delta^{(t(u))}|I(u)|)$, by Lemma 5.5.

By the algorithm, $I(u) \supseteq L(u)$, so summing these costs gives $cost_{I(u)}^{(t(u))} = \Omega(\delta^{(t(u))}|L(u)||S_K|)$, which is $\Omega(\delta^{(t(u))}|L(u)||\Delta^-(u)|)$ by Equation (20). □

Lemma 5.14. *Given a node $u \in \mathcal{F}$, $cost_{I(u)}^{(t(u))} = \Omega(\delta^{(t(u))}|C(u)||\Delta^+(u)|)$.*

Proof. Let S_J, S_{J^+}, S_R be defined as in Equation (10). It holds that $\Delta^+(u) = S_J \cup S_{J^+} \cup S_R$ by Equation (11), and these three sets are pairwise disjoint by definition. By Equation (14), the size of S_{J^+} is small compared to $|S_J|$, which implies that $|\Delta^+(u)| = O(|S_J \cup S_R|)$. Furthermore, by Equation (12), for any $i \in C(u), j \in S_J \cup S_R$, we have that their LP-distance is small, that is, $x_{i,j}^{(t)} < 0.6$.

We fix such a $j \in S_J \cup S_R$; therefore, $j \notin C(u)$. If $j \in S_J \cup S_R$ is not in the input-cluster $I(u)$, then $cost_{i,j}^{(t(u))} = \delta^{(t(u))}(1 - x_{i,j}^{(t(u))}) > \delta^{(t(u))}(1 - 0.6)$ for each $\{i, j\}$ pair with $i \in C(u)$. Else, $j \in I(u)$, but $j \notin L(u)$. That is because j is not in $C(u)$, and if it was in $L(u)$, then it would contradict Corollary 5.10. Therefore j was removed from the input-cluster in the LP-Cleaning step (Line 4 of Algorithm 2), and $cost_j^{(t(u))} = \Omega(\delta^{(t(u))}|I(u)|)$ by Lemma 5.5. Summing these costs proves our claim. □

5.5 Approximation Factor

In this section, we prove that Algorithm 1 is an $O(1)$ approximation of the LP cost.

We first make some definitions. Let $t \in [\ell]$. An input-cluster $C_I \in Q^{(t)}$ is strong if there exists a level-t node $u \in \mathcal{F}$ such that $I(u) = C_I$. Similarly, a part P of the output partition $P^{(t)}$ is strong if there exists a level-t node $u \in \mathcal{F}$ such that $C^+(u) = P$. In both cases, we say that u is the corresponding node. We characterize an input-cluster as weak if it is not strong, and similarly a part of the output partition $P^{(t)}$ as weak if it is not strong.

We start with upper bounding the cost of Algorithm 1. The upper bound is related to the input-clusters (distinguishing between strong and weak) and the parts of the output partitions (again distinguishing between strong and weak). Informally, for weak input-clusters and weak parts, the cost of our algorithm is proportional to the sum of squares of their size. For a strong input-cluster with corresponding node u, the cost of our algorithm is proportional to its size times the number of species of the input-cluster that did not end up in u's core-cluster. For a strong part of the output partitions, the cost of our algorithm is proportional to its size times the number of its species that did not end up in u's core-cluster.

Lemma 5.15. *Suppose we are given a Hierarchical Cluster Agreement instance $S, Q^{(*)}, \delta^{(*)}$ and LP-distances $x^{(*)}$. Then the cost of the output of Algorithm 1 at level t is at most*

$$\delta^{(t)}\left(\sum_{\substack{C_I \in Q^{(t)} \\ C_I \text{ is weak}}} \binom{|C_I|}{2} + \sum_{\substack{P \in P^{(t)} \\ P \text{ is weak}}} \binom{|P|}{2} + \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} \left(|I(u) \setminus C(u)||I(u)| + |\Delta^+(u)||C^+(u)|\right)\right).$$

Proof. The cost at level t is $\delta^{(t)}$ times the number of pairs $\{i,j\}$ that do not end up in the same part of the output partition $P^{(t)}$ but $\{i,j\} \in \mathcal{E}(Q^{(t)})$, plus the number of pairs $\{i,j\}$ that end up in the same part of the output partition but $\{i,j\} \notin \mathcal{E}(Q^{(t)})$. This is

$$\begin{aligned}
&\delta^{(t)}|\mathcal{E}(Q^{(t)}) \setminus \mathcal{E}(P^{(t)})| + \delta^{(t)}|\mathcal{E}(P^{(t)}) \setminus \mathcal{E}(Q^{(t)})| \\
&= \delta^{(t)} \sum_{C_I \in Q^{(t)}} \left|\binom{C_I}{2} \setminus \mathcal{E}(P^{(t)})\right| + \delta^{(t)} \sum_{P \in P^{(t)}} \left|\binom{P}{2} \setminus \mathcal{E}(Q^{(t)})\right| \\
&= \delta^{(t)} \sum_{\substack{C_I \in Q^{(t)} \\ C_I \text{ is weak}}} \left|\binom{C_I}{2} \setminus \mathcal{E}(P^{(t)})\right| + \delta^{(t)} \sum_{\substack{C_I \in Q^{(t)} \\ C_I \text{ is strong}}} \left|\binom{C_I}{2} \setminus \mathcal{E}(P^{(t)})\right| \\
&\quad + \delta^{(t)} \sum_{\substack{P \in P^{(t)} \\ P \text{ is weak}}} \left|\binom{P}{2} \setminus \mathcal{E}(Q^{(t)})\right| + \delta^{(t)} \sum_{\substack{P \in P^{(t)} \\ P \text{ is strong}}} \left|\binom{P}{2} \setminus \mathcal{E}(Q^{(t)})\right|.
\end{aligned}$$

Notice that if i,j are in the same core-cluster $C(u)$ of some node u at level $t(u) = t$, then $\{i,j\} \in \mathcal{E}(Q^{(t)}) \cap \mathcal{E}(P^{(t)})$. Also, for each strong input-cluster there exists a corresponding node u and vice versa (similarly for strong parts of the output-partitions). Therefore,

$$\begin{aligned}
\delta^{(t)} \sum_{\substack{C_I \in Q^{(t)} \\ C_I \text{ is strong}}} \left|\binom{C_I}{2} \setminus \mathcal{E}(P^{(t)})\right| &\le \delta^{(t)} \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} \binom{|I(u)|}{2} - \binom{|C(u)|}{2} \\
\delta^{(t)} \sum_{\substack{P \in P^{(t)} \\ P \text{ is strong}}} \left|\binom{P}{2} \setminus \mathcal{E}(Q^{(t)})\right| &\le \delta^{(t)} \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} \binom{|C^+(u)|}{2} - \binom{|C(u)|}{2}.
\end{aligned}$$

For an input-cluster C_I with a corresponding node u at level t such that $I(u) = C_I$, and since always $C(u) \subseteq I(u)$,

$$\binom{|I(u)|}{2} - \binom{|C(u)|}{2} = |I(u) \setminus C(u)||I(u)| - \binom{|I(u) \setminus C(u)|}{2} \le |I(u) \setminus C(u)||I(u)|.$$

Notice that subtraction is needed, since $|I(u) \setminus C(u)||I(u)|$ double-counts the pairs in $\binom{I(u)\setminus C(u)}{2}$.

Similarly, for a part P of the output partition $P^{(t)}$ with a corresponding node u at level t such that $C^+(u) = P$, and since always $C(u) \subseteq C^+(u)$,

$$\begin{aligned}
\binom{|C^+(u)|}{2} - \binom{|C(u)|}{2} = |C^+(u) \setminus C(u)||C^+(u)| - \binom{|C^+(u) \setminus C(u)|}{2} &\le |C^+(u) \setminus C(u)||C^+(u)| \\
&= |\Delta^+(u)||C^+(u)|.
\end{aligned}$$

□

For each term of Lemma 5.15, we show a matching lower bound for the LP cost. First, we give a lower bound of the LP cost related to the weak input-clusters.

Lemma 5.16. *The LP cost at level t $cost^{(t)}$ is*

$$\Omega\left(\delta^{(t)} \sum_{\substack{C_I \in Q^{(t)} \\ C_I \text{ is weak}}} \binom{|C_I|}{2}\right).$$

Proof. Let $C_I \in Q^{(t)}$ be an input-cluster and C_{LP} be the corresponding LP-cluster by Algorithm 2. By Lemma 5.5, $cost^{(t)}_{C_I \setminus C_{LP}} = \Omega(\delta^{(t)}|C_I \setminus C_{LP}||C_I|)$.

If C_I has no corresponding node u with $I(u) = C_I, t(u) = t$, then this means that $|C_{LP}| < 0.9|C_I|$, and therefore $|C_I \setminus C_{LP}| = \Omega(|C_I|)$, which makes the aforementioned cost $\Omega(\delta^{(t)}|C_I|^2) = \Omega(\delta^{(t)}\binom{|C_I|}{2})$.

Summing over all these input-clusters may only double-count each pair, which completes the proof. □

Next, we give a lower bound of the LP cost related to the strong input-clusters.

Lemma 5.17. *The LP cost at level t $cost^{(t)}$ is*

$$\Omega\left(\delta^{(t)} \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} (|I(u) \setminus C(u)||I(u)|)\right).$$

Proof. Summing the cost of Lemma 5.13 over all nodes u at level $t(u) = t$ gives a cost of

$$\Omega\left(\delta^{(t)} \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} (|L(u) \setminus C(u)||L(u)|)\right),$$

since we may only double-count some pairs. Similarly, summing the cost of Lemma 5.5 over all species in such nodes gives a cost of

$$\Omega\left(\delta^{(t)} \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} (|I(u) \setminus L(u)||I(u)|)\right).$$

We prove our claim by summing these two and noticing that $|L(u)| \geq 0.9|I(u)|$ by Line 5 of Algorithm 2. □

The following lemma lower bounds the LP cost in relation to the strong parts of the output partition $P^{(t)}$.

Lemma 5.18. *The LP cost at level t $cost^{(t)}$ is*

$$\Omega\left(\delta^{(t)} \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} \left(|\Delta^+(u)||C^+(u)|\right)\right).$$

Proof. Summing the cost of Lemma 5.14 over all nodes u at level $t(u) = t$ proves our claim, since we may only double-count some pairs. □

The following lemma lower bounds the LP cost in relation to the weak parts of the output partition $P^{(t)}$.

Lemma 5.19. *The LP cost at level t $cost^{(t)}$ is*

$$\Omega\left(\delta^{(t)} \sum_{\substack{P \in P^{(t)} \\ P \text{ is weak}}} \binom{|P|}{2}\right).$$

Proof. Fix any such part P. By Algorithm 3, each part of the output-partition $P^{(t)}$ corresponds to the extended-cluster of some node $u \in \mathcal{F}$. We use C_P to refer to the core-cluster of this node corresponding to P, and notice that $|C_P| = \Omega(|P|)$ by Equation (9). Furthermore, by Lemma 5.1 any two species $i, j \in C_P \subseteq P$ have $x_{i,j}^{(t)} < 0.2$.

We take two cases based on whether there exists an input-cluster $C_I \in Q^{(t)}$ such that $|C_I \cap C_P| > |C_P/2|$. Since $Q^{(t)}$ is a partition, there may be at most one such C_I for each part P. If none exists, then there exist $\Omega(\binom{|C_P|}{2})$ pairs of species in C_P that belong in different input-clusters, and thus $cost_{C_P}^{(t)} = \Omega(\delta^{(t)}\binom{|C_P|}{2}) = \Omega(\delta^{(t)}\binom{|P|}{2})$ by Equation (9).

For the remaining parts, we first partition them based on parts that have the same corresponding input-cluster. Let $P_1, \ldots, P_k$ be such a maximal group with the same corresponding input-cluster C_I, meaning that $|C_I| = \Omega(\sum_{r=1}^{k} |C_{P_r}|) = \Omega(\sum_{r=1}^{k} |P_r|)$. If C_I does not correspond to any node u at level t such that $I(u) = C_I$, then $cost_{C_I}^{(t)} = \Omega(\delta^{(t)}\binom{|C_I|}{2})$ by Lemma 5.5, which is $\Omega(\delta^{(t)} \sum_{r=1}^{k} \binom{|P_r|}{2})$.

Else there exists such a node u with $I(u) = C_I$, while by the statement of our Lemma there is no v such that $C^+(v) = P_r$ for $r \in [k]$. Therefore, all these parts are disjoint from $C^+(u)$ (Corollary 5.9) and thus disjoint from $C(u)$. This implies

$$\bigcup_{r=1}^{k} C_{P_r} \cap I(u) \subseteq I(u) \setminus C(u).$$

Then

$$|I(u) \setminus C(u)| \geq \left|\bigcup_{r=1}^{k} C_{P_r} \cap I(u)\right| > \sum_{r=1}^{k} |C_{P_r}|/2 = \Omega\left(\sum_{r=1}^{k} |P_r|\right).$$

By Lemma 5.17 the LP cost at level t is $\Omega(\delta^{(t)}|I(u) \setminus C(u)||I(u)|) = \Omega(\delta^{(t)} \sum_{r=1}^{k} \binom{|P_r|}{2})$. □

We are now ready to prove the following lemma:

Lemma 5.20. *Given a Hierarchical Cluster Agreement instance $S, Q^{(*)}, \delta^{(*)}$ and LP-distances $x^{(*)}$, the output of Derive-Hierarchy(S,LP-Cleaning($S, Q^{(*)}, x^{(*)}$)) is a sequence of hierarchical partitions $P^{(*)}$ with cost $O(cost^{(*)})$.*

Proof. By Corollary 5.9, the output is a sequence of hierarchical partitions of S.

For any level t, by Lemma 5.15 the cost of Derive-Hierarchy(S,LP-Cleaning($S, Q^{(*)}, x^{(*)}$)) is at most

$$\delta^{(t)}\left(\sum_{\substack{C_I \in Q^{(t)} \\ C_I \text{ is weak}}} \binom{|C_I|}{2} + \sum_{\substack{P \in P^{(t)} \\ P \text{ is weak}}} \binom{|P|}{2} + \sum_{\substack{u \in \mathcal{F} \\ t(u)=t}} \left(|I(u) \setminus C(u)||I(u)| + |\Delta^+(u)||C^+(u)|\right)\right).$$

By summing the LP cost at level t by Lemmas 5.16, 5.17, 5.18, and 5.19, we have that the LP cost at level t is

$$\Omega\left(\delta^{(t)}\left(\sum_{\substack{C_I\in Q^{(t)}\\ C_I \text{ is weak}}}\binom{|C_I|}{2}+\sum_{\substack{P\in P^{(t)}\\ P \text{ is weak}}}\binom{|P|}{2}+\sum_{\substack{u\in\mathcal{F}\\ t(u)=t}}\left(|I(u)\setminus C(u)||I(u)|+|\Delta^+(u)||C^+(u)|\right)\right)\right).$$

We conclude that the cost of the output of Derive-Hierarchy(S,LP-Cleaning($S, Q^{(*)}, x^{(*)}$)) is within a constant factor from the LP cost. Summing over all levels t proves our lemma. □

Lemma 5.20 directly proves

$$\text{HierClustAgree} = O(1), \qquad \text{(E) from Figure 1}$$

as Algorithm 1 simply picks $x^{(*)}$ to be an optimal fractional solution to the LP relaxation.

Combining this with Inequality C concludes the proof of Theorem 3.1.

6 CONSTANT INTEGRALITY GAP

In this section, we prove that the LP formulation for Hierarchical Correlation Clustering (Section 2) has constant integrality gap. This directly extends to the integrality gap of the LP formulation used by Ailon and Charikar for ultrametrics [3], as the LP formulation for Hierarchical Correlation Clustering is a generalization of the one for ultrametrics (implicit in References [3, 40], discussed in Section 7).

Notice that this is not direct from our algorithm, as for Hierarchical Correlation Clustering we do not directly work with the LP from Section 2; we rather reduce our problem to an instance of Hierarchical Cluster Agreementand then round the LP of this instance.

We start with some definitions. Suppose we have an instance of Hierarchical Correlation Clustering $S, \delta^{(*)} = (\delta^{(1)}, \dots, \delta^{(\ell)}), E^{(*)} = (E^{(1)}, \dots, E^{(\ell)})$. We say that x is an LP vector if it consists of LP distances $x_{i,j}$ satisfying the triangle inequality and being in the interval $[0, 1]$ for all species $i, j \in S$. For any $E \subseteq \binom{S}{2}$, we extend the previously used notion of LP cost as

$$cost^{(t)}(E, x) = \delta^{(t)}\left(\sum_{\{i,j\}\in E}(x_{i,j}) + \sum_{\{i,j\}\notin E}(1 - x_{i,j})\right).$$

Notice that for any LP vector x and edge-sets $E_1, E_2 \subseteq \binom{S}{2}$, we have that

$$cost^{(t)}(E_1, x) \le cost^{(t)}(E_2, x) + \delta^{(t)}|E_1 \triangle E_2|. \tag{22}$$

That is because only pairs in the symmetric difference may be charged differently by $cost^{(t)}(E_1, x)$ and $cost^{(t)}(E_2, x)$, and the maximum such difference is $\delta^{(t)}$, as the LP-distances are between 0 and 1.

The LP formulation of Correlation Clustering, which is a special case of the formulation of Hierarchical Correlation Clustering, has constant integrality gap [17]. Therefore, for a Correlation Clustering instance S, E, integral solution Q whose cost is within a constant factor from the optimal integral solution OPT, and any t and LP vector x, it holds that

$$\delta^{(t)}|E \triangle Q| = O(\delta^{(t)}|E \triangle OPT|) = O(cost^{(t)}(E, x)). \tag{23}$$

Finally, let $Q^{(*)} = (Q^{(1)}, \dots, Q^{(\ell)})$ be partitions of S such that for each $t \in [\ell]$, $Q^{(t)}$ is a solution to Correlation Clustering with input $S, E^{(t)}$ whose cost is within a constant factor from the optimal. Let $x^{(*)} = (x^{(1)}, \dots, x^{(\ell)})$ be ℓ LP vectors satisfying Equation (6) that are an optimal fractional solution to Hierarchical Correlation Clustering.

We need to prove that some integral solution $P^{(*)} = (P^{(1)}, \dots, P^{(l)})$ to Hierarchical Correlation Clustering has cost within a constant factor of the optimal fractional solution. We pick $P^{(*)} = \text{Derive-Hierarchy}(S, \text{LP-Cleaning}(S, Q^{(*)}, x^{(*)}))$, that is, the integral solution suggested by Lemma 5.20. Formally, we prove

$$\sum_{t=1}^{\ell} \delta^{(t)} |P^{(t)} \triangle E^{(t)}| = O\left(\sum_{t=1}^{\ell} cost^{(t)}(E^{(t)}, x^{(*)})\right).$$

It holds by the triangle inequality that

$$\sum_{t=1}^{\ell} \delta^{(t)} |P^{(t)} \triangle E^{(t)}| \leq \sum_{t=1}^{\ell} \delta^{(t)} (|P^{(t)} \triangle \mathcal{E}(Q^{(t)})| + |\mathcal{E}(Q^{(t)}) \triangle E^{(t)}|).$$

By Lemma 5.20, we have that

$$\begin{aligned}\sum_{t=1}^{\ell} \delta^{(t)} |P^{(t)} \triangle \mathcal{E}(Q^{(t)})| &= O\left(\sum_{t=1}^{\ell} cost^{(t)}(\mathcal{E}(Q^{(t)}), x^{(*)})\right) \\ &\leq O\left(\sum_{t=1}^{\ell} \left(cost^{(t)}(E^{(t)}, x^{(*)}) + \delta^{(t)} |E^{(t)} \triangle \mathcal{E}(Q^{(t)})|\right)\right),\end{aligned}$$

with the latter inequality following by Equation (22). Therefore, we bound $\sum_{t=1}^{\ell} \delta^{(t)} |P^{(t)} \triangle E^{(t)}|$:

$$\begin{aligned}\sum_{t=1}^{\ell} \delta^{(t)} |P^{(t)} \triangle E^{(t)}| &\leq \sum_{t=1}^{\ell} \delta^{(t)} (|P^{(t)} \triangle \mathcal{E}(Q^{(t)})| + |\mathcal{E}(Q^{(t)}) \triangle E^{(t)}|) \\ &= O\left(\sum_{t=1}^{\ell} \left(cost^{(t)}(E^{(t)}, x^{(*)}) + \delta^{(t)} |E^{(t)} \triangle \mathcal{E}(Q^{(t)})|\right)\right) \\ &= O\left(\sum_{t=1}^{\ell} cost^{(t)}(E^{(t)}, x^{(*)})\right),\end{aligned}$$

with the last step following from Equation (23).

7 FROM L_1-FITTING ULTRAMETRICS TO HIERARCHICAL CORRELATION CLUSTERING

For completeness, we here review the reduction from ultrametrics to hierarchical correlation clustering implicit in previous work [3, 40].

Given an L_1-fitting ultrametrics instance with input $\mathcal{D} : \binom{S}{2} \rightarrow \mathbb{R}_{>0}$, we construct an input to the Hierarchical Correlation Clustering instance as follows. Let $D^{(1)} < \dots < D^{(\ell+1)}$ be the distinct distances that appear in the input distance function $\mathcal{D}$. For $t = 1, \dots, \ell$, define

$$\delta^{(t)} = D^{(t+1)} - D^{(t)} \text{ and } E^{(t)} = \left\{ \{i, j\} \in \binom{S}{2} \mid \mathcal{D}(i, j) \leq D^{(t)} \right\}. \tag{24}$$

Now given the solution to this hierarchical correlation clustering problem, to construct a corresponding ultrametric tree, we first complete the partition hierarchy with $P^{(0)}$ partitioning S into singletons and $P^{(\ell+1)}$ consisting of the single set S. Moreover, we set $\delta^{(0)} = D^{(1)}$.

To get the ultrametric tree U, we create a node for each set in the hierarchical partitioning. Next, for $t = 0, \dots, \ell$, the parent of a level t node u is the node on level $t+1$ whose set contains the set of u, and the length of the parent edge is $\delta^{(t)}/2$. Then nodes on level t are of height $\sum_{i=0}^{t-1} \delta^{(t)}/2 = D^{(t)}/2$, and if two species have their lowest common ancestor on level t, then their distance is exactly $D^{(t)}$.

The construction is reversible in a manner that given any ultrametric tree U with leaf set S and all distances from $\{D^{(1)}, \dots, D^{(\ell+1)}\}$, we get the partitions $P^{(1)}, \dots, P^{(\ell)}$ as follows. First, possibly by introducing nodes with only one child, for each species i we make sure it has ancestors of heights $D^{(t)}/2$ for $t = 1, \dots, \ell + 1$. Then, for $t = 1, \dots, \ell$, we let partitions $P^{(t)}$ consist of the sets of descendants for each level t node in U.

With this relation between U and $P^{(1)}, \dots, P^{(\ell)}$, it follows easily that they have the same cost relative to $\mathcal{D}$ in the sense that

$$\sum_{t=1}^{\ell} \delta^{(t)} |E^{(t)} \,\Delta\, E(P^{(t)})| = \sum_{\{i,j\} \in \binom{S}{2}} |dist_U(i,j) - \mathcal{D}(i,j)|.$$

Thus, with Equation (24), the hierarchical correlation clustering is equal to L_1-fitting ultrametrics with ultrametric distances from the set of different distances in $\mathcal{D}$.

Finally, from Lemma 1(a) in Reference [40], we have that among all ultrametrics minimizing the L_1 distance to $\mathcal{D}$, there is at least one using only distances from $\mathcal{D}$. This implies that an α-approximation algorithm for hierarchical correlation clustering implies an α-approximation algorithm for L_1-fitting ultrametrics (but not the other way around, as hierarchical correlation clustering is a more general problem). Therefore,

$$\text{UltraMetric} \leq \text{HierCorrClust}. \qquad \text{(B) from Figure 1}$$

Combining this with Theorem 3.1 concludes the second part of Theorem 1.1, namely that the L_1-fitting ultrametrics problem can be solved in deterministic polynomial time within a constant approximation factor.

8 TREE METRIC TO ULTRAMETRIC

Agarwala et al. [2] reduced tree metrics to certain restricted ultrametrics. In fact, their reduction may make certain species have distance 0 in the final tree, which means that it is actually a reduction from tree pseudometrics[7] to certain restricted ultrametrics. In this section, we show that the restrictions are not needed and that the reduction can be made in a way that does not introduce zero-distances. While none of this is hard, note that for completeness we have to deal with these issues for L_1 to claim the results presented in this article. Here we get a clean black-box reduction for all $L_p, p \geq 1$. Thus, in the future, if, say, we get a constant factor approximation for ultrametrics in L_2, then we can immediately claim a constant factor for tree metrics in L_2 as well.

8.1 Reducing the Tree Pseudometric Problem to the (Unrestricted) Ultrametric Problem

The claim from Reference [2] is that approximating a certain restricted ultrametric within a factor α can be used to approximate tree pseudometric within a factor 3α. Here we completely lift these restrictions for L_1 and show that they can be lifted for all L_p with $p \in \{2, 3, \dots\}$ with an extra factor of at most 2.

We will need the well-known characterization of ultrametrics discussed in the Introduction that U is an ultrametric iff it is a metric and $\forall \{i,j,k\} \in \binom{S}{3} : U(i,j) \leq \max\{U(i,k), U(k,j)\}$.

For simplicity, we prove the theorem only for $p < \infty$, as for L_∞ the 3 approximation [2] cannot be improved by our theorem.

[7]Pseudometrics are a generalization of metrics that allow distance 0 between distinct species.

Theorem 8.1. *For any integer $1 \le p < \infty$, a factor $\alpha \ge 1$ approximation for L_p-fitting ultrametrics implies a factor $3 \cdot 2^{(p-1)/p} \cdot \alpha$ approximation for L_p-fitting tree pseudometrics. In particular, for L_1 it implies a factor 3α.*

Proof (Extending proof from [2]). The restriction from Agarwala et al. [2] is as follows. For every species $i \in S$, we have a "lower bound" β_i. Moreover, we have a distinguished species $\kappa \in S$ with an upper bound γ_κ.

We want an ultrametric U such that

$$\gamma_\kappa \ge U(i,j) \ge \max\{\beta_i, \beta_j\} \qquad \forall \{i,j\} \in \binom{S}{2}.$$
$$\beta_\kappa = \gamma_\kappa$$

We note that the conditions can only be satisfied if $\gamma_\kappa \ge \beta_i$ for all $i \in S$, so we assume this is the case.

The result from Reference [2] states that for any p and $\mathcal{D} : \binom{S}{2} \to \mathbb{R}_{>0}$, if we can minimize the restricted ultrametric L_p error within a factor α in polynomial-time, then there is a polynomial-time algorithm that minimizes the tree pseudometric L_p error within a factor 3α.

We start with creating a new distance function $\mathcal{D}'$,

$$\mathcal{D}'(i,j) = \min\{\gamma_\kappa, \max\{\mathcal{D}(i,j), \beta_i, \beta_j\}\}.$$

Intuitively, we squeeze $\mathcal{D}'$ to satisfy the restrictions. For any restricted ultrametric U, the error between U and $\mathcal{D}'$ can never be larger than the error between U and $\mathcal{D}$, no matter the norm L_p. Formally, since U is restricted, we have $\max\{\beta_i, \beta_j\} \le U(i,j) \le \gamma_\kappa$,

- If $\mathcal{D}(i,j) > \gamma_\kappa$, then $\mathcal{D}'(i,j) = \gamma_\kappa \ge U(i,j)$ and $|U(i,j) - \mathcal{D}'(i,j)|^p < |U(i,j) - \mathcal{D}(i,j)|^p$.
- If $\mathcal{D}(i,j) < \max\{\beta_i, \beta_j\}$, then $\mathcal{D}'(i,j) = \max\{\beta_i, \beta_j\} \le U(i,j)$ and $|U(i,j) - \mathcal{D}'(i,j)|^p < |U(i,j) - \mathcal{D}(i,j)|^p$.
- If $\max\{\beta_i, \beta_j\} \le \mathcal{D}(i,j) \le \gamma_\kappa$, then $\mathcal{D}'(i,j) = \mathcal{D}(i,j)$ and $|U(i,j) - \mathcal{D}'(i,j)|^p = |U(i,j) - \mathcal{D}(i,j)|^p$.

We now ask for an arbitrary ultrametric fit U' for $\mathcal{D}'$. With exactly the same reasoning, we can only improve the cost if we replace U' with

$$U(i,j) = \min\{\gamma_\kappa, \max\{U'(i,j), \beta_i, \beta_j)\}\}.$$

Clearly, U now satisfies the restrictions (in the end of the proof we show that it is an ultrametric).

Our solution to L_p-fitting tree pseudometrics is to first create $\mathcal{D}'$ from $\mathcal{D}$, obtain ultrametric U' by an α approximation to L_p-fitting ultrametrics, and then obtain the restricted ultrametric U from U'. Finally, we apply the result from Reference [2] to get the tree pseudometric.

Let $OPT_{\mathcal{D},R}$ be the closest restricted ultrametric to $\mathcal{D}$ and $OPT_{\mathcal{D}'}$ be the closest ultrametric to $\mathcal{D}'$. It suffices to show that $\|U - \mathcal{D}\|_p \le 2^{(p-1)/p}\alpha\|OPT_{\mathcal{D},R} - \mathcal{D}\|_p$ (equivalently $\|U - \mathcal{D}\|_p^p \le 2^{p-1}\alpha^p\|OPT_{\mathcal{D},R} - \mathcal{D}\|_p^p$) and that U is indeed an ultrametric.

By the above observations, it holds that

$$\|\mathcal{D}' - U\|_p \le \|\mathcal{D}' - U'\|_p \le \alpha\|\mathcal{D}' - OPT_{\mathcal{D}'}\|_p \le \alpha\|\mathcal{D}' - OPT_{\mathcal{D},R}\|_p \implies$$
$$\|\mathcal{D}' - U\|_p^p \le \alpha^p\|\mathcal{D}' - OPT_{\mathcal{D},R}\|_p^p.$$

By definition of $\mathcal{D}'$, and since U is restricted, for any species i,j it holds $\min\{\mathcal{D}(i,j), U(i,j)\} \le \mathcal{D}'(i,j) \le \max\{\mathcal{D}(i,j), U(i,j)\}$. The proof follows by a direct case study of the three cases $\mathcal{D}(i,j) \le \max\{\beta_i, \beta_j\}$, $\max\{\beta_i, \beta_j\} < \mathcal{D}(i,j) \le \gamma_\kappa$, and $\gamma_\kappa < \mathcal{D}(i,j)$. Therefore,

$$|\mathcal{D}(i,j) - U(i,j)| = |\mathcal{D}(i,j) - \mathcal{D}'(i,j)| + |\mathcal{D}'(i,j) - U(i,j)|.$$

For $p \geq 1$, we have $|x|^p + |y|^p \leq (|x| + |y|)^p$, meaning $|\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + |\mathcal{D}'(i,j) - U(i,j)|^p \leq |\mathcal{D}(i,j) - U(i,j)|^p$.

Moreover, by the convexity of $|x|^p$ for real x, we get $((x+y)/2)^p \leq (|x|^p + |y|^p)/2$, meaning $|\mathcal{D}(i,j) - U(i,j)|^p \leq 2^{p-1}(|\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + |\mathcal{D}'(i,j) - U(i,j)|^p)$. Therefore,

$$\begin{aligned}|\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + |\mathcal{D}'(i,j) - U(i,j)|^p &\leq |\mathcal{D}(i,j) - U(i,j)|^p \\ &\leq 2^{p-1}(|\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + |\mathcal{D}'(i,j) - U(i,j)|^p).\end{aligned}$$

The same holds if we replace U with $OPT_{\mathcal{D},R}$, as we only used that U is restricted. We now have

$$\begin{aligned}\|\mathcal{D} - U\|_p^p &= \sum_{\{i,j\}\in\binom{S}{2}} |\mathcal{D}(i,j) - U(i,j)|^p \\ &\leq \sum_{\{i,j\}\in\binom{S}{2}} 2^{p-1}(|\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + |\mathcal{D}'(i,j) - U(i,j)|^p) \\ &= 2^{p-1}\left(\sum_{\{i,j\}\in\binom{S}{2}} |\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + \|\mathcal{D}' - U\|_p^p\right) \\ &\leq 2^{p-1}\left(\sum_{\{i,j\}\in\binom{S}{2}} |\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + \alpha^p\|\mathcal{D}' - OPT_{\mathcal{D},R}\|_p^p\right) \\ &\leq 2^{p-1}\alpha^p\left(\sum_{\{i,j\}\in\binom{S}{2}} |\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + \|\mathcal{D}' - OPT_{\mathcal{D},R}\|_p^p\right) \\ &= 2^{p-1}\alpha^p \sum_{\{i,j\}\in\binom{S}{2}} (|\mathcal{D}(i,j) - \mathcal{D}'(i,j)|^p + |\mathcal{D}'(i,j) - OPT_{\mathcal{D},R}(i,j)|^p) \\ &\leq 2^{p-1}\alpha^p \sum_{\{i,j\}\in\binom{S}{2}} (|\mathcal{D}(i,j) - OPT_{\mathcal{D},R}(i,j)|^p) \\ &= 2^{p-1}\alpha^p\|\mathcal{D} - OPT_{D,R}\|_p^p.\end{aligned}$$

Finally, we need to prove that U inherits that it is an ultrametric. This is clear if we proceed in rounds; in each round we construct a new ultrametric, and the last one will coincide with U.

More formally, let $U_0 = U'$. In the first $|S|$ rounds, we take out a different $i' \in S$ at a time and let

$$U_r(i',j) = \max\{U_{r-1}(i',j), \beta_{i'}\} \qquad \forall j \neq i.$$

Suppose $r > 0$ is the first round where U_r is not an ultrametric. Then there exists a triple $\{i,j,k\}$ such that $U_r(i,j) > \max\{U_r(i,k), U_r(k,j)\}$. As we only increase distances, this may only happen if $U_r(i,j) > U_{r-1}(i,j)$. But this means that at round r we picked either i or j (w.l.o.g. assume it was i) and set $U_r(i,j) = \beta_i$. However, this would also give $U_r(i,k) \geq \beta_i = U_r(i,j)$, contradicting $U_r(i,j) > \max\{U_r(i,k), U_r(k,j)\}$.

Finally, U is simply

$$U(i,j) = \min\{\gamma_\kappa, U_{|S|}(i,j)\}.$$

Suppose there exists a triple $\{i,j,k\}$ that now violates the ultrametric property; then it holds that

$$U(i,j) > \max\{U(i,k), U(k,j)\}.$$

As we did not increase any distance, this means that both $U(i,k) < U_{|S|}(i,k)$ and $U(k,j) < U_{|S|}(k,j)$; but distances can only reduce to γ_κ, which is an upper bound on $U(i,j)$ by construction. □

8.2 From Tree Metric to Tree Pseudometric

In this section, we prove that to find a good tree metric, it suffices to find a good tree pseudometric. This is a minor detail that we add for completeness. Informally, the construction simply replaces 0 distances with some parameter ϵ and accordingly adapts the whole metric. By making the parameter ϵ very small, the cost is not significantly changed.

Technically, our main lemma is the following.

Lemma 8.2. *Given is a set S, a distance function $\mathcal{D} : \binom{S}{2} \to \mathbb{R}_{>0}$, a tree T with non-negative edge weights describing a tree pseudometric on S, and a parameter $\alpha \in (0,1]$. In time polynomial in the size of T we can construct a tree T' with positive edge weights describing a tree metric on S, such that for any $p \geq 1$, it holds that $\|T' - \mathcal{D}\|_p \leq (1+\alpha)\|T - \mathcal{D}\|_p$.*

Proof. We construct T' from T as follows. First, we contract all edges with weight 0. This may result in several species from S coinciding in the same node. For each such node u and species i coinciding with some other species in u, we create a new leaf-node u_i connected only with u with edge-weight $\epsilon > 0$ (to be specified later). We identify i with u_i, instead of u.

T' describes a tree metric on S, as by construction each species $i \in S$ is identified with a distinct node in T', and T' only contains positive edge-weights.

If T matches $\mathcal{D}$ exactly, that is, $\|T - \mathcal{D}\|_p = 0$, then no pair of species $i,j \in S$ have $dist_T(i,j) = 0$, as $\mathcal{D}(i,j) > 0$. But then no species coincided in the same node due to the contractions, meaning that no distances changed, which proves our claim. From here onward we assume that at least one pair has $dist_T(i,j) \neq \mathcal{D}(i,j)$.

To specify the parameter ϵ we first make some definitions. Let Y be the set containing all species $i \in S$ for which we created a new leaf node in T'. Moreover, let d_{min} be the smallest positive $|dist_T(i,j) - \mathcal{D}(i,j)|$ among all $i,j \in S$. Then,

$$\epsilon = \alpha d_{min}/(8|S|).$$

For any two species i,j, their distance stays the same, increases by ϵ, or increases by 2ϵ. Therefore, for $p = \infty$ we directly get $\|T' - \mathcal{D}\|_p \leq \|T - \mathcal{D}\|_p + 2\epsilon$. By definition of d_{min}, we also have $\|T - \mathcal{D}\|_p \geq d_{min} \implies 2\epsilon \leq \alpha\|T - \mathcal{D}\|_p/(4|S|) < \alpha\|T - \mathcal{D}\|_p$, which proves our claim. Therefore, we can assume that $p < \infty$.

We start with a lower bound related to $\|T - \mathcal{D}\|_p$. By definition of Y, for any $i \in Y$ there exists a $j \in Y$ such that $dist_T(i,j) = 0$, meaning that $|dist_T(i,j) - \mathcal{D}(i,j)| = |\mathcal{D}(i,j)| \geq d_{min}$. Therefore,

$$\|T - \mathcal{D}\|_p^p \geq \frac{|Y|}{2} d_{min}^p.$$

We now upper bound $\|T' - \mathcal{D}\|_p$. If $dist_T(i,j) \neq \mathcal{D}(i,j)$, then $|dist_T(i,j) - \mathcal{D}(i,j)| \geq d_{min}$ by definition of d_{min}. For the rest of the pairs i,j, if their distance increased, then either $i \in Y$ or $j \in Y$ by construction; thus, there are at most $|Y||S|$ such pairs. Using these observations, we take the following three cases (where the first one uses $\epsilon = \alpha d_{min}/(8|S|)$):

$$\begin{aligned}
\sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)\neq\mathcal{D}(i,j)}} |dist_{T'}(i,j)-\mathcal{D}(i,j)|^p &\leq \sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)\neq\mathcal{D}(i,j)}} (|dist_T(i,j)-\mathcal{D}(i,j)|+2\epsilon)^p \\
&\leq \sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)\neq\mathcal{D}(i,j)}} \left(|dist_T(i,j)-\mathcal{D}(i,j)|+\frac{\alpha}{4}d_{min}\right)^p \\
&\leq \sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)\neq\mathcal{D}(i,j)}} \left(1+\frac{\alpha}{4}\right)^p |dist_T(i,j)-\mathcal{D}(i,j)|^p
\end{aligned}$$

$$\sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)=\mathcal{D}(i,j) \\ dist_T(i,j)=dist_{T'}(i,j)}} |dist_{T'}(i,j)-\mathcal{D}(i,j)|^p = 0$$

$$\sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)=\mathcal{D}(i,j) \\ dist_T(i,j)<dist_{T'}(i,j)}} |dist_{T'}(i,j)-\mathcal{D}(i,j)|^p \leq \sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)=\mathcal{D}(i,j) \\ dist_T(i,j)<dist_{T'}(i,j)}} |2\epsilon|^p \leq |Y||S||2\epsilon|^p.$$

Adding these three upper bounds $\|T'-\mathcal{D}\|_p^p$ by

$$\sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)\neq\mathcal{D}(i,j)}} \left(1+\frac{\alpha}{4}\right)^p |dist_T(i,j)-\mathcal{D}(i,j)|^p + |Y||S||2\epsilon|^p.$$

Using our lower bound and the definition of ϵ,

$$|Y||S||2\epsilon|^p = |Y||S|\left(\frac{\alpha d_{min}}{4|S|}\right)^p < \frac{|Y|}{2}\left(\frac{\alpha}{2}\right)^p d_{min}^p \leq \left(\frac{\alpha}{2}\right)^p \|T-\mathcal{D}\|_p^p.$$

Therefore, we get

$$\begin{aligned}
\|T'-\mathcal{D}\|_p^p &\leq \sum_{\substack{\{i,j\}\in\binom{S}{2} \\ dist_T(i,j)\neq\mathcal{D}(i,j)}} \left(1+\frac{\alpha}{4}\right)^p |dist_T(i,j)-\mathcal{D}(i,j)|^p + (\alpha/2)^p\|T-\mathcal{D}\|_p^p \\
&\leq \left(1+\frac{\alpha}{4}\right)^p \|T-\mathcal{D}\|_p^p + (\alpha/2)^p\|T-\mathcal{D}\|_p^p \\
&< (1+\alpha)^p\|T-\mathcal{D}\|_p^p
\end{aligned}$$

and thus $\|T'-\mathcal{D}\|_p \leq (1+\alpha)\|T-\mathcal{D}\|_p$. □

Therefore, for any $p \geq 1$, we can approximate L_p-fitting tree metrics by using an approximation to L_p-fitting tree pseudometrics. The error is at most $(1+\alpha)$ times the approximation factor of the tree pseudometric, as any tree metric is also a tree pseudometric.

Setting $\alpha = \frac{1}{|S|}$ and using the result from Reference [2], we conclude that

$$\text{TreeMetric} \leq (3+o(1)) \cdot \text{UltraMetric}. \qquad \text{(A) from Figure 1}$$

This concludes the proof of Theorem 1.1.

As a final note, in the case of L_0 (that is, we count the number of disagreements between $\mathcal{D}$ and T') one cannot hope for a similar result. To see this, let S be a set of species, and let $c_1, c_2 \in S$ be two special species. The distance between any pair of species is 2, except if the pair contains either

c_1 or c_2, in which case the distance is 1. The optimal tree pseudometric simply sets the distance between c_1 and c_2 to 0 and preserves everything else (1 disagreement).

Any tree metric requires at least $|S|-3$ disagreements: We say that a non-special species is good if it has tree-distance 1 to both c_1 and c_2 and bad otherwise. Bad species have distance different than 1 to at least one special species, while good species have distance less than 2 with each other; the disagreements minimize at $|S| - 3$, when there is either one or two good species.

9 APX-HARDNESS

The problems of L_1-fitting tree metrics and L_1-fitting ultrametrics are regarded as APX-Hard in the literature [3, 40]. However, we decided to include our own versions of these proofs for a multitude of reasons: First and foremost, Reference [3] attributes the APX-hardness to Reference [53], which is an unpublished Master thesis that is non-trivial to read. Also Reference [40] claims that APX-Hardness of L_1-fitting ultrametrics follows directly by the APX-Hardness of Correlation Clustering [17], but this is only true if all the distances in the ultrametric are in $\{1, 2\}$. Second, we think that our proofs are considerably simpler and more direct. Finally, our constant factor approximation algorithms for these problems make it important to have formal proofs of their APX-Hardness, since the combination settles that a constant factor approximation is best possible in polynomial time unless P=NP.

9.1 L_1-Fitting Ultrametrics

The correlation clustering problem has been shown to be APX-Hard in Reference [17]. As noted in References [3, 40] correlation clustering is the same as the L_1-fitting ultrametrics in case both the input and the output are only allowed to have distances in $\{1, 2\}$. We refer to this problem as L_1-fitting $\{1, 2\}$-ultrametrics. Therefore, the L_1-fitting $\{1, 2\}$-ultrametrics is also APX-Hard.

For completeness, we sketch this relation here. Let $E \subseteq \binom{S}{2}$ be an instance of correlation clustering; then $\mathcal{D}(i, j)$ is an instance to L_1-fitting $\{1, 2\}$-ultrametrics, where $\mathcal{D}(i, j) = 1$ if $\{i, j\} \in E$, and $\mathcal{D}(i, j) = 2$ otherwise. Similarly, given $\mathcal{D}$ we can obtain E by setting $\{i, j\} \in S$ iff $\mathcal{D}(i, j) = 1$. Given any solution to correlation clustering (permutation P of S), we get a solution T to L_1-fitting $\{1, 2\}$-ultrametrics with $T(i, j) = 1$ if i, j are in the same part of P and $T(i, j) = 2$ otherwise. As T is an ultrametric, we are guaranteed that $T(i, j) \le \max\{T(i, k), T(j, k)\}$; therefore, if $T(i, k) = T(j, k) = 1$, then $T(i, j) = 1$ as only distances in $\{1, 2\}$ are allowed. Thus distance-1 is a transitive relation and P can be obtained by the equivalence classes of species with distance 1 in T. The observation from Reference [3] is that $|E \triangle \mathcal{E}(P)| = \|T - \mathcal{D}\|_1$, which follows by trivial calculations.

The bird's eye view of our approach for showing APX-Hardness of L_1-fitting ultrametrics is the following. For the sake of contradiction, we assume that L_1-fitting ultrametrics is not APX-Hard. We then show how to solve the L_1-fitting $\{1, 2\}$-ultrametrics problem in polynomial time within any constant factor greater than 1, contradicting the fact that it is APX-Hard. The main idea is that we first solve the general L_1-fitting ultrametrics problem. Then we apply a sequence of local transformations that converts the general ultrametric to an ultrametric with distances in $\{1, 2\}$ without increasing the error. To achieve this, we first eliminate distances smaller than 1, then eliminate distances larger than 2, and then eliminate distances in $(1, 2)$.

We first prove the following result concerning the local transformations. We remind the reader that an ultrametric T is defined as a metric with the property that for $i, j, k \in S$ we have $T(i, j) \le \max\{T(i, k), T(j, k)\}$.

We note that the ideas in Lemma 1 of Reference [40] can be directly used to prove Lemma 9.1. We only include Lemma 9.1, because we need these ideas applied to any ultrametric, not just to optimal ones, as Lemma 1 of Reference [40] does.

Lemma 9.1. *Let S be a set of species, $\mathcal{D} : \binom{S}{2} \to \{1,2\}$ be a distance function with distances only in $\{1,2\}$, and T be a rooted tree such that each species $i \in S$ corresponds to a leaf in T (more than one species may correspond to the same leaf) and all leaves are at the same depth. Then, in polynomial time, we can create a tree $T_{1,2}$ describing an ultrametric with distances only in $\{1,2\}$ such that $\|T_{1,2} - \mathcal{D}\|_1 \leq \|T - \mathcal{D}\|_1$.*

Proof. We set $T' = T$ and apply the following local transformation to T'. If $T(i,j) < 1$, then we set $T'(i,j) = 1$. It holds that $\|T' - \mathcal{D}\|_1 \leq \|T - \mathcal{D}\|_1$ as $\mathcal{D}(i,j) \geq 1$ and $T(i,j) < 1$ implies $|1 - \mathcal{D}(i,j)| < |T(i,j) - \mathcal{D}(i,j)|$. Furthermore, T' still describes an ultrametric. To see this, notice that $\max\{T'(i,k), T'(j,k)\} \geq \max\{T(i,k), T(j,k)\} \geq T(i,j)$ as we do not decrease distances and T is an ultrametric. Therefore, if $T'(i,j) > \max\{T'(i,k), T'(j,k)\}$, then this means that $T'(i,j) > T(i,j)$. But this only happens if $T'(i,j) = 1$, which is a lower bound on $T'(i,k), T'(j,k)$ by construction. This contradicts that $T'(i,j) > \max\{T'(i,k), T'(j,k)\}$, and therefore T' describes an ultrametric. Notice that no two species in S coincide in the same node in T' as the minimum distance between any two distinct species is 1.

Similarly, we set $T'' = T'$ and apply the following local transformation to T''. If $T'(i,j) > 2$, then we set $T''(i,j) = 2$. It holds that $\|T'' - \mathcal{D}\|_1 \leq \|T' - \mathcal{D}\|_1$ as $\mathcal{D}(i,j) \leq 2$ and $T'(i,j) > 2$ imply $|2 - \mathcal{D}(i,j)| < |T'(i,j) - \mathcal{D}(i,j)|$. Furthermore T'' still describes an ultrametric. To see this, notice that $T''(i,j) \leq T'(i,j) \leq \max\{T'(i,k), T'(j,k)\}$. If $T''(i,j) > \max\{T''(i,k), T''(j,k)\}$, then $\max\{T''(i,k), T''(j,k)\} < \max\{T'(i,k), T'(j,k)\}$, which only happens if either of $T'(i,k)$ or $T'(j,k)$ dropped to 2, meaning that $\max\{T''(i,k), T''(j,k)\} = 2$. But 2 is an upper bound on $T''(i,j)$. This contradicts that $T''(i,j) > \max\{T''(i,k), T''(j,k)\}$, and therefore T'' describes an ultrametric.

Now, by construction, the ultrametric tree describing T'' has leaves at depth 1 (the maximum distance is 2) and internal nodes at depth between 0 and 0.5 (the minimum distance is 1). If an internal node u has depth $d_u \in (0, 0.5)$, then let x_1 be the number of pairs $\{i,j\} \subseteq \binom{S}{2}$ whose nearest common ancestor is u and $\mathcal{D}(i,j) = 1$ and x_2 be the number of pairs $\{i,j\} \subseteq \binom{S}{2}$ whose nearest common ancestor is u and $\mathcal{D}(i,j) = 2$. If $x_2 \geq x_1$, then we remove u and connect the children of u directly with the parent of u. We still have an ultrametric, as we have an ultrametric tree describing the metric. The L_1 error is not larger, as the error of x_2 pairs drops by twice the absolute difference in depths between u and its parent (their distance increases but does not exceed 2), and the error of $x_1 \leq x_2$ pairs increases by the same amount; otherwise, $x_2 < x_1$. In this case, we increase the depth of u until it coincides with the depth of some of its children and merge these children with u. Similarly with the previous argument, we still have an ultrametric with smaller L_1 error.

Each time we apply the above step, we remove at least one node from our tree. Therefore, when we can no longer apply this step, we spent polynomial time and acquired an ultrametric $T_{1,2}$ with distances only in $\{1,2\}$ whose L_1 error from $\mathcal{D}$ is $\|T_{1,2} - \mathcal{D}\|_1 \leq \|T'' - \mathcal{D}\|_1 \leq \|T' - \mathcal{D}\|_1 \leq \|T - \mathcal{D}\|_1$. □

Theorem 9.2. *L_1-fitting ultrametrics is APX-Hard. In particular, L_1-fitting ultrametrics where the input only contains distances in $\{1,2\}$ is APX-Hard.*

Proof. Let $\mathcal{D} : \binom{S}{2} \to \{1,2\}$ be a distance function, OPT be the optimal ultrametric for the L_1-fitting ultrametrics problem, and $OPT_{1,2}$ be the optimal ultrametric for the L_1-fitting $\{1,2\}$-ultrametrics. We solve this L_1-fitting ultrametrics instance in polynomial time and obtain T such that $\|T - \mathcal{D}\|_1 \leq (1+\epsilon)OPT$ for a sufficiently small constant ϵ, as we assumed that L_1-fitting ultrametrics is not APX-Hard. Notice that any solution to the L_1-fitting $\{1,2\}$-ultrametrics is also a solution to the L_1-fitting ultrametrics, meaning that $\|T - \mathcal{D}\|_1 \leq (1+\epsilon)OPT \leq (1+\epsilon)OPT_{1,2}$.

Let $T_{1,2}$ be the ultrametric we get from T by applying Lemma 9.1. Then $T_{1,2}$ is a solution to the L_1-fitting $\{1,2\}$-ultrametrics instance, and $\|T_{1,2} - \mathcal{D}\|_1 \leq \|T - \mathcal{D}\|_1 \leq (1+\epsilon)OPT_{1,2}$. This contradicts the fact that L_1-fitting $\{1,2\}$-ultrametrics is APX-Hard. □

9.2 L_1-Fitting Tree Metrics

In this section, we show that L_1-fitting tree metrics is APX-Hard. Our reduction is based on the techniques used in Reference [28] to prove NP-Hardness of the same problem. The bird's eye view of our approach is that we solve L_1-fitting ultrametrics by solving L_1-fitting tree metrics on a modified instance. In this instance, we introduce new species having small distance to each other and large distance to the original species. Through a sequence of local transformations, we show that we can modify the tree describing the obtained tree metric so as to consist of a star connecting the new species and an ultrametric tree connecting the original species (the center of the star and the root of the ultrametric tree are connected by a large edge). This ultrametric would refute APX-Hardness of L_1-fitting ultrametrics in case L_1-fitting tree metrics was not APX-Hard.

Theorem 9.3. *L_1-fitting tree metrics is APX-Hard.*

Proof. Let $\mathcal{D} : \binom{S}{2} \to \{1,2\}$ be an input to L_1-fitting ultrametrics, such that all distances in $\mathcal{D}$ are in $\{1,2\}$. Moreover, let $n = |S|$ and $OPT_{\mathcal{D},U}$ be the ultrametric minimizing $\|OPT_{\mathcal{D},U} - \mathcal{D}\|_1$. By Theorem 9.2, this problem is APX-Hard. For the sake of contradiction, assume L_1-fitting tree metrics is not APX-Hard.

Let $\epsilon \in (0,1)$ be a sufficiently small constant and $M = 2(1+\epsilon)\binom{n}{2} + 1$. We extend S to $S' \supseteq S$ such that $|S'| = 2n$. For $\{i,j\} \in \binom{S}{2}$, we set $\mathcal{D}'(i,j) = \mathcal{D}(i,j)$. For $i,j \in \binom{S' \setminus S}{2}$, we set $\mathcal{D}'(i,j) = 2$. For all other i,j we set $\mathcal{D}'(i,j) = M$. As we assumed L_1-fitting tree metrics not to be APX-Hard, in polynomial time we can compute T, a tree metric such that for any other tree metric T_0 it holds that $\|T - \mathcal{D}'\|_1 \leq (1+\epsilon)\|T_0 - \mathcal{D}'\|$ for sufficiently small ϵ such that $0 < \epsilon < 1$.

We first show that each species $k \in S' \setminus S$ has an incident edge contained in all paths from this species to any species in S. To do so, we need to upper bound $\|T - \mathcal{D}'\|_1$. If we make a star whose leaves are the species in S with distance 1 from the center, a second star whose leaves are the species in $S' \setminus S$ with distance 1 from the center, and connect the two centers with an edge of weight $M - 2$, then only pairs with both species in S may have the wrong distance, and the error for each such pair is at most 1. Therefore, $\|T - \mathcal{D}'\|_1 \leq (1+\epsilon)\binom{n}{2}$. This means that if $k \in S' \setminus S$, then in the tree describing T there exists a path Π_k starting from k and having weight larger than 1, such that the path from k to any species $i \in S$ has Π_k as a prefix. To see why this is true, notice that otherwise two species i,j would exist such that the paths from k to i and from k to j only share a prefix $\Pi_{i,j}$ of weight $w_{\Pi_{i,j}} \leq 1$. But $T(i,k) > M/2$, as otherwise we would have $\|T - \mathcal{D}'\|_1 \geq |T(i,k) - \mathcal{D}'(i,k)| \geq M/2 > (1+\epsilon)\binom{n}{2}$ and similarly $T(j,k) > M/2$. Then $T(i,j) = T(i,k) + T(j,k) - 2 \cdot w_{\Pi_{i,j}} > M - 2$, meaning, again, $\|T - \mathcal{D}'\|_1 > |T(i,j) - \mathcal{D}'(i,j)| > (1+\epsilon)\binom{n}{2}$.

Using the aforementioned structural property, we show how to modify our tree so that all species in S are close to each other, all species in $S' \setminus S$ are close to each other, but species in S are far from species in $S' \setminus S$. Let $k \in S' \setminus S$ be the species minimizing $\sum_{i \in S} |T(i,k) - \mathcal{D}'(i,k)|$. We transform the tree describing T by inserting a node u in the path Π_k at distance 1 from k and creating a star with u as its center and all species in $S' \setminus S$ as leaves at distance 1. Let T' be the resulting tree metric and notice that $\|T' - \mathcal{D}'\|_1 \leq \|T - \mathcal{D}'\|_1$, because the errors from species in $S' \setminus S$ to species in S did not increase (by definition of k), the errors between species in $S' \setminus S$ are exactly zero, and the errors between species in S stay exactly the same (we did not modify the part of the tree formed by the union of paths between species in S).

Then, we modify the tree describing T' to obtain T'' so that the distance from any species in S to any species in $S' \setminus S$ is M. If for any $i \in S$ we have $T'(i,k) \neq M$, then we move i in the tree so as to make its distance with k equal to M: If $T'(i,k) < M$, then we create a new leaf node connected with i with distance $M - T'(i,k)$ and move i to this new leaf node. Else if $T'(i,k) > M$, then there exists an i' (possibly by subdividing an edge) in the path from k to i having distance M from k, and we move i to this node. Notice that $\|T'' - \mathcal{D}'\|_1 \leq \|T' - \mathcal{D}'\|_1$, because we move each $i \in S$ by $|M - T'(i,k)|$ so that it has zero error with each $k' \in S' \setminus S$, meaning that the error drops by $|S' \setminus S||M - T'(i,k)| = n|M - T'(i,k)|$ ($|M - T'(i,k)|$ for each $k' \in S' \setminus S$) and increases by at most $(n-1)|M - T'(i,k)|$ ($|M - T'(i,k)|$ for each $i' \in S \setminus \{i\}$).

If we remove all nodes not in a path from k to any $i \in S$ in the tree describing T'', then by construction we have a tree $T_{\mathcal{D},U}$ rooted at k, having leaves identified with the species in S and all leaves having depth M. By the above discussion, its error is $\|T_{\mathcal{D},U} - \mathcal{D}\|_1 = \|T'' - \mathcal{D}'\|_1$. As some species may coincide in the same nodes, we get an ultrametric $T'_{\mathcal{D},U}$ of S having the aforementioned properties so that no two species coincide in the same node, using Lemma 9.1.

Notice that $OPT_{\mathcal{D},U}$ has maximum distance between species less than M; otherwise, its error would be at least $M-2$, which is a contradiction to the fact that an ultrametric where all species have distance 1 has error at most $\binom{n}{2} < M-2$. But then we can take the tree describing this optimal ultrametric, connect its root with a node u so that u has distance $M-1$ to all species in S, and identify each species $k' \in S' \setminus S$ with a leaf $u'_{k'}$ connected with u with an edge of weight 1. If the resulting tree metric is T_1, then $\|OPT_{\mathcal{D},U} - \mathcal{D}\|_1 = \|T_1 - \mathcal{D}'\|_1$. We conclude that $\|T'_{\mathcal{D},U} - \mathcal{D}\|_1 = \|T_{\mathcal{D},U} - \mathcal{D}\|_1 = \|T'' - \mathcal{D}'\|_1 \leq \|T' - \mathcal{D}'\|_1 \leq \|T - \mathcal{D}'\|_1 \leq (1+\epsilon)\|T_1 - \mathcal{D}'\|_1 = (1+\epsilon)\|OPT_{\mathcal{D},U} - \mathcal{D}\|_1$. This contradicts Theorem 9.2. □

10 CONCLUSION

We have given, to the best of our knowledge, the first constant factor approximation for L_1-fitting tree metrics, the first improvement on the problem since the late 2000s. This problem was one of the relatively few remaining problems for which obtaining a constant factor approximation or showing hardness was open. Breaking through the best-known $O((\log n)(\log\log n))$-approximation had thus been stated as a fascinating open problem.

In this article, we set the parameters related to our algorithm in a way that makes the presentation clearer. Given the current non-optimized parameters, it can be verified that the approximation factor for Hierarchical Cluster Agreement is less than 400, for Hierarchical Correlation Clustering (and thus also for L_1-fitting ultrametrics) it is less than 1,600, and for L_1-fitting tree metrics it is less than 4,800.

Interestingly, our journey brought us to the study of a natural definition of hierarchical cluster agreement that may be of broader interest, in particular to the data mining community where correlation clustering has been a successful objective function and where hierarchical clustering is often desired in practice.

Finding a polynomial time constant factor approximation (or showing that this is hard, e.g., by reduction to unique games) for L_2-fitting tree metrics is a great open problem. Recall from Section 8 that it suffices to focus on approximating the problem of fitting into an arbitrary ultrametric (no need for restricted versions). Finally, the $O((\log n)(\log\log n))$-approximation algorithm of Ailon and Charikar for the weighted case (where the cost of an edge is weighted by an input edge weight) could potentially be improved to $O(\log n)$ without improving multicut, and it would be interesting to do so. Going even further would require improving the best-known bounds for multicut, a notoriously hard problem.

REFERENCES

[1] Amir Abboud, Vincent Cohen-Addad, and Hussein Houdrouge. 2019. Subquadratic high-dimensional hierarchical clustering. In *NeurIPS*. 11576–11586.

[2] Richa Agarwala, Vineet Bafna, Martin Farach, Mike Paterson, and Mikkel Thorup. 1999. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM J. Comput.* 28, 3 (1999), 1073–1085.

[3] Nir Ailon and Moses Charikar. 2011. Fitting tree metrics: Hierarchical clustering and phylogeny. *SIAM J. Comput.* 40, 5 (2011), 1275–1291.

[4] Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: Ranking and clustering. *J. ACM* 55, 5 (2008), 23:1–23:27.

[5] Noga Alon, Yossi Azar, and Danny Vainstein. 2020. Hierarchical clustering: A 0.585 revenue approximation. In *COLT*, Vol. 125. 153–162.

[6] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. 2008. A discriminative framework for clustering via similarity functions. In *STOC*. 671–680.

[7] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Mach. Learn.* 56, 1-3 (2004), 89–113.

[8] Yair Bartal. 1996. Probabilistic approximations of metric spaces and its algorithmic applications. In *FOCS*. 184–193.

[9] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. 2014. An improved approximation for k-median, and positive correlation in budgeted optimization. In *SODA*. 737–756.

[10] Gunnar E. Carlsson and Facundo Mémoli. 2010. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.* 11, 47 (Apr. 2010), 1425–1470.

[11] L. L. Cavalli-Sforza and A. W. F. Edwards. 1967. Phylogenetic analysis models and estimation procedures. *Am. J. Hum. Genet.* 19, 3 (1967), 233–257.

[12] James A. Cavender. 1978. Taxonomy with confidence. *Math. Biosci.* 40, 3 (1978), 271–280.

[13] Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. 2020. From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering. In *NeurIPS*.

[14] Moses Charikar and Vaggos Chatziafratis. 2017. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *SODA*. 841–854.

[15] Moses Charikar, Vaggos Chatziafratis, and Rad Niazadeh. 2019. Hierarchical clustering better than average-linkage. In *SODA*. 2291–2304.

[16] Moses Charikar, Vaggos Chatziafratis, Rad Niazadeh, and Grigory Yaroslavtsev. 2019. Hierarchical clustering for euclidean data. In *AISTATS*. 2721–2730.

[17] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. 2005. Clustering with qualitative information. *J. Comput. Syst. Sci.* 71, 3 (2005), 360–383. Announced at FOCS 2003.

[18] Vaggos Chatziafratis, Grigory Yaroslavtsev, Euiwoong Lee, Konstantin Makarychev, Sara Ahmadian, Alessandro Epasto, and Mohammad Mahdian. 2020. Bisect and conquer: Hierarchical clustering via max-uncut bisection. In *AISTATS*, Vol. 108. 3121–3132.

[19] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. 2006. On the hardness of approximating multicut and sparsest-cut. *Comput. Complex.* 15, 2 (2006), 94–114. Announced at CCC 2005.

[20] Shuchi Chawla, Konstantin Makarychev, Tselil Schramm, and Grigory Yaroslavtsev. 2015. Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *STOC*. 219–228.

[21] Michael Cochez and Hao Mou. 2015. Twister tries: Approximate hierarchical agglomerative clustering for average distance in linear time. In *SIGMOD*. 505–517.

[22] Vincent Cohen-Addad, Rémi de Joannis de Verclos, and Guillaume Lagarde. 2021. Improving ultrametrics embeddings through coresets. In *ICML*.

[23] Vincent Cohen-Addad, Varun Kanade, and Frederik Mallmann-Trenn. 2017. Hierarchical clustering beyond the worst-case. In *NeurIPS*. 6201–6209.

[24] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. 2019. Hierarchical clustering: Objective functions and algorithms. *J. ACM* 66, 4 (2019), 26:1–26:42. Announced at SODA 2018.

[25] Vincent Cohen-Addad, Karthik C. S., and Guillaume Lagarde. 2020. On efficient low distortion ultrametric embedding. In *ICML*, Vol. 119. 2078–2088.

[26] Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. 2022. Correlation clustering with sherali-adams. *To appear FOCS'22* (2022), 651–661.

[27] Sanjoy Dasgupta. 2016. A cost function for similarity-based hierarchical clustering. In *STOC*. 118–127.

[28] William H. E. Day. 1987. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull. Math. Biol.* 49, 4 (1987), 461–467.

[29] Kedar Dhamdhere. 2004. Approximating additive distortion of embeddings into line metrics. In *APPROX-RANDOM*. 96–104.

[30] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2004. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.* 69, 3 (2004), 485–497.

[31] Chenglin Fan, Anna C. Gilbert, Benjamin Raichel, Rishi Sonthalia, and Gregory Van Buskirk. 2020. Generalized metric repair on graphs. In *SWAT*, Vol. 162. 25:1–25:22.

[32] Chenglin Fan, Benjamin Raichel, and Gregory Van Buskirk. 2018. Metric violation distance: Hardness and approximation. In *SODA*. SIAM, 196–209.
[33] Martin Farach and Sampath Kannan. 1999. Efficient algorithms for inverting evolution. *J. ACM* 46, 4 (1999), 437–449.
[34] Martin Farach, Sampath Kannan, and Tandy J. Warnow. 1995. A robust model for finding optimal evolutionary trees. *Algorithmica* 13, 1/2 (1995), 155–179.
[35] James S. Farris. 1972. Estimating phylogenetic trees from distance matrices. *Am. Natur.* 106, 951 (1972), 645–688.
[36] Anna C. Gilbert and Lalit Jain. 2017. If it ain't broke, don't fix it: Sparse metric repair. In *Allerton*. IEEE, 612–619.
[37] Anna C. Gilbert and Rishi Sonthalia. 2018. Unsupervised metric learning in presence of missing data. In *Allerton*. IEEE, 313–321.
[38] Teofilo F Gonzalez. 1985. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.* 38 (1985), 293–306. https://www.sciencedirect.com/science/article/pii/0304397585902245
[39] Sudipto Guha and Samir Khuller. 1999. Greedy strikes back: Improved facility location algorithms. *J. Algor.* 31, 1 (1999), 228–248. Announced at SODA 1998.
[40] Boulos Harb, Sampath Kannan, and Andrew McGregor. 2005. Approximating the best-fit tree under l_p norms. In *APPROX-RANDOM*. 123–133.
[41] Monika Rauch Henzinger, Valerie King, and Tandy J. Warnow. 1999. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica* 24, 1 (1999), 1–13.
[42] Jon M. Kleinberg and Éva Tardos. 2006. *Algorithm Design*. Addison-Wesley.
[43] Frank Thomson Leighton and Satish Rao. 1999. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM* 46, 6 (1999), 787–832.
[44] Bin Ma, Lusheng Wang, and Louxin Zhang. 1999. Fitting distances by tree metrics with increment error. *J. Combin. Optim.* 3, 2-3 (1999), 213–225.
[45] Benjamin Moseley and Joshua Wang. 2017. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *NeurIPS*. 3094–3103.
[46] Elchanan Mossel and Sébastien Roch. 2005. Learning nonsingular phylogenies and hidden markov models. In *STOC*. 366–375.
[47] Aurko Roy and Sebastian Pokutta. 2016. Hierarchical clustering via spreading metrics. In *NeurIPS*. 2316–2324.
[48] Anastasios Sidiropoulos, Dingkang Wang, and Yusu Wang. 2017. Metric embeddings with outliers. In *SODA*. 670–689.
[49] Peter H. A. Sneath and Robert R. Sokal. 1962. Numerical taxonomy. *Nature* 193, 4818 (1962), 855–860.
[50] Peter H. A. Sneath and Robert R. Sokal. 1963. *Numerical Taxonomy. The Principles and Practice of Numerical Classification*. Freeman.
[51] Rishi Sonthalia and Anna C. Gilbert. 2020. Tree! i am no tree! i am a low dimensional hyperbolic embedding. In *NeurIPS*.
[52] Anke van Zuylen and David P. Williamson. 2009. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Math. Operat. Res.* 34, 3 (2009), 594–620.
[53] Harold Todd Wareham. 1993. *On the Computational Complexity of Inferring Evolutionary Trees*. Master's thesis. Memorial University of of Newfoundland.
[54] M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. 1977. Additive evolutionary trees. *J. Theor. Biol.* 64, 2 (1977), 199–213.

Received 14 December 2021; revised 19 December 2023; accepted 21 December 2023