Triplet Reconstruction and all other Phylogenetic CSPs are Approximation Resistant

Vaggos Chatziafratis
Computer Science and Engineering
University of California at Santa Cruz
Santa Cruz, CA, USA
vaggos@ucsc.edu

Konstantin Makarychev
Department of Computer Science
Northwestern University
Evanston, IL, USA
konstantin@northwestern.edu

***Abstract*—We study the natural problem of Triplet Reconstruction (also known as Rooted Triplets Consistency or Triplet Clustering), originally motivated by applications in computational biology and relational databases (Aho, Sagiv, Szymanski, and Ullman, 1981) [2]: given n datapoints, we want to embed them onto the n leaves of a rooted binary tree (also known as a hierarchical clustering, or ultrametric embedding) such that a given set of m *triplet* constraints is satisfied. A triplet constraint $ij|k$ for points i, j, k indicates that "i, j are more closely related to each other than to k," (in terms of distances $d(i, j) \leq d(i, k)$ and $d(i, j) \leq d(j, k)$) and we say that a tree satisfies the triplet $ij|k$ if the distance in the tree between i, j is smaller than the distance between i, k (or j, k). Among all possible trees with n leaves, can we efficiently find one that satisfies a large fraction of the m given triplets?**

Aho et al. (1981) [2] studied the decision version and gave an elegant polynomial-time algorithm that determines whether or not there exists a tree that satisfies all of the m constraints. Moreover, it is straightforward to see that a random binary tree achieves a constant $\frac{1}{3}$-approximation, since there are only 3 distinct triplets $ij|k, ik|j, jk|i$ (each will be satisfied w.p. $\frac{1}{3}$). Unfortunately, despite more than four decades of research by various communities, there is no better approximation algorithm for this basic Triplet Reconstruction problem.

Our main theorem—which captures Triplet Reconstruction as a special case—is a general hardness of approximation result about Constraint Satisfaction Problems (CSPs) over *infinite* domains (CSPs where instead of boolean values $\{0, 1\}$ or a fixed-size domain, the variables can be mapped to any of the n leaves of a tree). Specifically, we prove that assuming the Unique Games Conjecture [57], Triplet Reconstruction and more generally, *every* Constraint Satisfaction Problem (CSP) over *hierarchies* is approximation resistant, i.e., there is no polynomial-time algorithm that does asymptotically better than a *biased* random assignment.

Our result settles the approximability not only for Triplet Reconstruction, but for many interesting problems that have been studied by various scientific communities such as the popular Quartet Reconstruction and Subtree/Supertree Aggregation Problems. More broadly, our result significantly extends the list of approximation resistant predicates by pointing to a large new family of hard problems over hierarchies. Our main theorem is a generalization of Guruswami, Håstad, Manokaran, Raghavendra, and Charikar (2011) [36], who showed that *ordering* CSPs (CSPs over permutations of n elements, e.g., Max Acyclic Subgraph, Betweenness, Non-Betweenness) are approximation resistant. The main challenge in our analyses stems from the fact that trees have *topology* (in contrast to permutations and ordering CSPs) and it is the tree topology that determines whether a given constraint on the variables is satisfied or not. As a byproduct, we also present some of the first CSPs where their approximation resistance is proved against biased random assignments, instead of uniformly random assignments.

***Index Terms*—Constraint Satisfaction, Approximation Resistance, Unique Games**

I. Introduction

The algorithmic task of constructing *hierarchical* representations of data has been studied by various communities over many decades with applications ranging from statistics [44], [82] and databases [2] to the analysis of complex networks, such as the Internet or social networks [29], [66], and more recently, to machine learning, where hierarchical embeddings have proven useful for understanding text, images, graphs and multi-relational data [63]. The reason why they are so ubiquitous is that many real data sets stemming from nature or society are organized according to a latent hierarchy [66]. Interestingly, many relevant questions and algorithmic ideas originated in the field of Taxonomy and Phylogenetics [31], [34], [71] with the goal of classifying all living and extinct organisms into the Tree of Life.

The easiest way to visualize such hierarchical representations for a given data set is by using a dendrogram, also known as a *Hierarchical Clustering*. Hierarchical clustering is an embedding of the data set to a tree, often depicted as a rooted binary tree whose leaves are in one-to-one correspondence with the points in the data set, see Figure 1. The hierarchical clustering tree shows the recursive partitioning of the data set into successively

Vaggos Chatziafratis was supported by a UCSC start-up grant. Part of this work was done while VC was a postdoc at Northwestern, and a FODSI fellow at MIT and Northeastern.

Konstantin Makarychev was supported by NSF Awards CCF-1955351, CCF-1934931, and EECS-2216970.

smaller and smaller clusters. Observe that all data points are clustered together at the root, but eventually they get separated at the leaves (internal nodes correspond to intermediate subclusters formed by descendant leaves).

In contrast to "flat" clustering techniques like k-means/k-median which cannot capture fine-grained relationships among points or groups of points, hierarchical clustering reveals the structure of a data set at multiple levels of granularity simultaneously. For example, consider *triplet* queries of the form "*Among 3 items i, j, k, which two are most closely related?*"; a quick look at the hierarchical clustering (see Fig. 1) immediately reveals the answer by locating the 3 leaves i, j, k and noticing which of the 3 gets separated first from the other two. Answering such triplet comparisons is easy for humans which makes them popular in metric learning and crowdsourcing settings [10], [32], [77], [80], and understanding how to accurately aggregate a large collection of such triplet queries into a hierarchical clustering was the primary motivation of our work. As we will see, studying triplets will lead us to interesting connections with hardness of approximation and approximation resistant predicates.

In this paper, we study the approximability of a large class of Constraint Satisfaction Problems (CSPs) over *hierarchies*, i.e., trees, which have been studied in various communities, including in databases [2], in logic and algebra [18], in computational biology [24], [34] and in theoretical computer science [5], [19], [28], [54], [72]. The input is a collection of (potentially inconsistent) local relationships between k items of a ground set (with total size n), and we are asked to find the hierarchical clustering that *maximizes* agreement with the input. Such local relationships can take the form of triplet or quartet constraints (or even quintuples etc.), and more generally, they can be a k-arity constraint on k data points which prescribes how the k data points should be split by the final hierarchy. For the most common examples of triplet and quartet constraints, please see Figure 1.

For readers familiar with Correlation Clustering [13], we should note here that it is different in at least three important ways: first, in correlation clustering the desired output is a (flat) partition of the data points (whereas in hierarchical clustering we want a mapping to the n leaves of a tree), second, constraints in correlation clustering are between pairs of points (whereas in hierarchical clustering the input specifies constraints on triplets, quartets etc.) and third, there are technical differences (as we show) in terms of their behavior with respect to approximation resistance.

A. Our Contributions

We revisit several old questions in Hierarchical Clustering and CSPs over *Infinite*-Domains and prove *tight* upper and lower bounds under the Unique Games Conjecture [57], thus settling the approximability of a large class of hierarchical reconstruction problems. Interestingly, we extend the notion of *approximation resistant* CSPs [41] to allow for *biased* random assignments (instead of uniform random assignments), and our main hardness result for CSPs over trees holds under this extended definition, which could be of independent interest. As far as we know, our results provide the first approximation resistant CSPs where the optimal approximation threshold is achieved by a *non-uniform* random assignment.

Recall that for CSPs over infinite domains, the variables are not boolean and instead of taking values $\{0, 1\}$ (or in a fixed-size domain), they are allowed to be mapped to infinite domains. Prominent examples include Correlation Clustering [13] and *Ordering CSPs*, i.e., CSPs over permutations of n elements, such as Max Acyclic Subgraph, Betweenness, Non-Betweenness [36]; for our case, the infinite domain corresponds to the n leaves of a hierarchical tree which of course grows as the number n of data points grows. In fact, our results generalize the hardness results of Guruswami et al. (2011) [36], since a permutation corresponds (in a formal sense) to a special case of hierarchical clustering (because we consider ordered trees). At a high-level, the main challenge in our problems comes from the fact that CSPs over trees depend on the tree's *topology* and whether a given constraint is satisfied or not is determined by how and in what order exactly various data points got split at intermediate nodes. Observe that this is irrelevant for correlation clustering and for permutations. Specifically, we settle the approximability of Triplet Reconstruction, Quartet Reconstruction, and General CSPs over Trees.

Triplet Reconstruction (also Rooted Triplets Consistency): Aggregating triplets into a hierarchical clustering was originally asked in the context of relational databases by Aho et al. (1981) [2]. A triplet constraint $ij|k$ indicates that "items i, j are more similar to each other than to k." Given m triplets, we would like to construct a hierarchical clustering on the n items, that satisfies as many constraints as possible, i.e., k is split first from i, j (see also Fig. 1).

Quartet Reconstruction: When constraints are on 4 points a, b, c, d, they are called "quartet" constraints. The task is to find a (rooted or unrooted) tree that satisfies as many of the quartet constraints as possible (Fig. 1). A special case of Quartet Reconstruction is the popular "Unrooted Quartet Consistency" problem in computational biology [5], [15], [34], [54], [72], [73], [75].

General CSPs over Trees: The previous two problems are only two special cases of general CSPs over

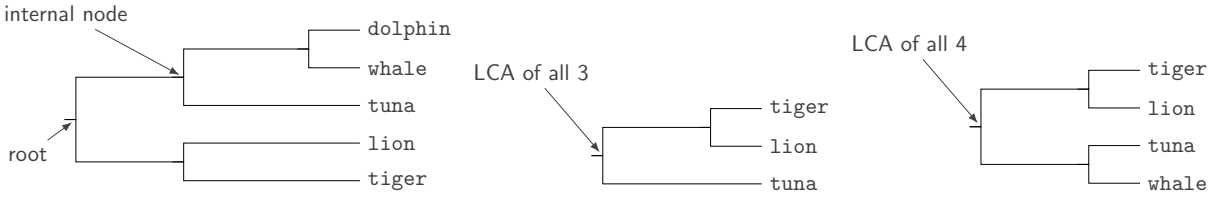

Fig. 1. Hierarchical Clustering on 5 points (Left), a triplet constraint (Middle) and a quartet constraint (Right) satisfied by hierarchical clustering. The internal node shown corresponds to the subcluster {whale, dolphin, tuna}. The basic constituent of a hierarchy is a *triplet comparison* or *triplet constraint*, e.g., {{lion, tiger}|{tuna}} indicates the closest pair among the 3. Formally, the Lowest Common Ancestor (LCA) of {lion, tiger} is a descendant of the LCA of all 3. Another type of a more complicated comparison is among 4 points, e.g., the *quartet* comparison {{lion, tiger}|{tuna, whale}} prescribes what's the correct split. We can see that the hierarchical clustering satisfies the shown triplet and quartet constraint (it also satisfies triplets {{whale, dolphin}|{tuna}}, {{whale, dolphin}|{lion}}, {{whale, tuna}|{tiger}}, and quartets {{whale, dolphin}|{lion, tiger}}, {{{whale, dolphin}|{tuna}}|{lion}}.

trees. Specifically, Triplet Reconstruction is a CSP of arity 3 and Quartet Reconstruction is a CSP of arity 4. However, there is no reason to stop there; in fact, the algebraic and logic communities have extensively studied what happens if we allow for trees with larger fan-out degree, or for conjunctions (logical $\wedge$) or disjunctions (logical $\vee$) between constraints,[1] or for the constraints to be of arity k. In the algebraic and logic literature [16], [18], such CSPs are termed *Phylogenetic CSPs* due to their connections to popular "Consensus Tree" or "Subtree/Supertree Aggregation" methods in computational biology [1], [53], [61], [68], [75].

Before stating our general theorem, let us focus only on our first result about the Triplet Reconstruction problem and highlight its status in terms of approximability and hardness. Then, it will be much easier to understand our results for Quartet Reconstruction and for General CSPs over Trees (along with the main technical challenges).

B. Result I: Beating Random is Hard for Triplet Reconstruction

We will need the following simple definitions (for examples, see Fig 1):

Definition I.1 (Triplet)**.** *A triplet t, denoted $t = ab|c$, is a rooted, unordered, binary tree on 3 leaves a, b, c. A rooted, unordered, binary tree T (containing leaves a, b, c) is said to be* consistent with t *(or T* satisfies t*), if the LCA(a, b) in T is a proper descendant of LCA(a, c) in T. Otherwise, the triplet and the tree are* inconsistent with each other *(or T* violates t*). In general, triplets can also have weights* $\text{weight}(ab|c)$.

The natural optimization problem associated with Triplet Reconstruction is MAXTRIPLETS:

[1]For example, $ij|k \vee ik|j$ captures the *forbidden* triplets problem, that forbids triplets $jk|i$ from the final hierarchy.

Definition I.2 (MAXTRIPLETS Problem)**.** *Given a set X of n data points and m triplets defined on data points from X, find the hierarchical clustering (i.e., the binary rooted tree) that is consistent with as many triplets as possible (per the definition above).*

MAXTRIPLETS is NP-hard in general instances, but Aho, Sagiv, Szymanski, and Ullman (1981) [2] presented an algorithm for completely satisfiable instances of MAXTRIPLETS: This algorithm finds a tree that is consistent with all given triplets, if such tree exists, otherwise it halts and declares that the triplets are conflicting and no tree can satisfy all of the triplets.

The following trivial algorithm achieves a $\frac{1}{3}$-approximation: "output a *uniformly random* tree on the n data points." Observe that for 3 items a, b, c, there are only 3 distinct triplets—namely $ab|c, ac|b, bc|a$—and so with probability $\frac{1}{3}$, the uniformly random tree will satisfy each of the input triplets. Surprisingly, despite four decades of research, this is currently the best known approximation for triplet reconstruction. Our first result shows that being stuck at the trivial $\frac{1}{3}$-approximation ratio is not a coincidence:

Theorem I.3. *For every constant $\varepsilon > 0$, it is UG-hard to distinguish instances of* MAXTRIPLETS*, where a $(1 - \varepsilon)$ fraction of the triplets can be satisfied by a hierarchical clustering, from* MAXTRIPLETS *instances where at most a $(\frac{1}{3} + \varepsilon)$ fraction can be satisfied.*

Stated simply, we prove that if ρ is the expected fraction of constraints satisfied by a uniformly random tree, then obtaining a ρ'-approximation for any constant $\rho' > \rho$ is UG-hard. In other words, we show that MAXTRIPLETS is *approximation resistant*. Recall that a predicate is approximation resistant if it is NP-hard (or UG-hard in our case) to approximate the corresponding

CSP significantly better than what is achieved by the trivial algorithm that picks an assignment uniformly at random. For example, 3SAT is approximation resistant [41] and so is every ordering CSP such as Max Acyclic Subgraph, Betweenness, Non-Betweenness [36]. Prior to our work, the best known hardness of approximation for MAXTRIPLETS was 2/3 due to Chatziafratis, Mahdian, and Ahmadian (2021) [27].

C. Result II: From Triplets to Hardness of General CSPs over Trees

Given our first result on the hardness of MAXTRIPLETS, it is natural to wonder what happens in terms of approximability if we increase the arity of the constraints from 3 to 4, i.e., what happens for Quartet Reconstruction and its associated optimization versions of MAXQUARTETS (we defer the exact definitions for now, but hopefully the problem is clear). For MAXQUARTETS even though there are results [54] that give a PTAS for very dense instances (density here implies that there is a quartet for every four data points, thus $m = \Omega(n^4)$), the approximability in the general case remained open: *How well can we approximate* MAXQUARTETS *in polynomial time?* Again, for the most well-studied case of unrooted quartet reconstruction [15], [34], [54], a trivial algorithm that outputs an unrooted tree uniformly at random is a constant approximation, and this has been the state-of-the-art in the worst-case for many decades. In light of our Theorem I.3, we are able to settle the approximability for MAXQUARTETS, proving that this trivial algorithm is again optimal (under UGC) (see Appendix C).

a) General CSPs over Trees.: Triplets and Quartets are two special cases of a more general family of CSPs over trees that are not well-understood from a theoretical perspective. Such general CSPs over trees are also studied in the algebraic and logic communities under the name *Phylogenetic CSPs* [16]–[18], which will be borrowed here.[2] For formal definitions, please see Preliminaries (Section III).

After seeing our hardness results for MAXTRIPLETS and MAXQUARTERS, one may assume that the random assignment algorithm always gives the best possible approximation (ignoring $o(1)$ terms). However, as we discuss in Section IV, this is not the case. In fact, for some phylogenetic CSPs the uniform random assignment algorithm satisfies exponentially small in k fraction of all constraints while other algorithms satisfy e.g., a constant (not depending on k) fraction of all constraints. It turns out that the best algorithms for arbitrary phylogenetic CSPs are *biased* random assignment algorithms. We show the following result.

> **Theorem I.4** (Informal)**.** *Assuming the UGC, every Phylogenetic CSP is approximation resistant. Interestingly, this holds for a more general notion of approximation resistance, where* ***biased*** *random solutions are allowed (not just uniformly random outputs like in boolean CSPs and ordering CSPs).*

b) On Approximation Resistance.: The subject of approximation resistance is a fascinating topic in computation with a rich literature, and despite the intensive efforts to characterize the approximability of CSPs, it is not yet clear what properties characterize them in general. It is perhaps striking, but many CSPs are approximation resistant, and two fundamental examples are MAX3SAT, MAX3LIN [41]. In contrast, for arity 2, Håstad (2005) [42] showed that no predicate that depends on two inputs (e.g., MAXCUT) from an arbitrary finite domain can be approximation resistant. Investigating higher arity CSPs has also yielded interesting results: for arity 3, a precise classification of approximation resistant 3CSPs is known [83], but for arity 4 and higher the situation is unclear [8], [9], [40], [43]. For example, Hast gave a characterization for 355 out of 400 different predicate types for binary 4CSPs. Moreover, Håstad (2007) [43] showed, under UGC, that a random k-ary predicate for large k is approximation resistant. More recently, Guruswami and Lee (2015) [38] showed hardness for the family of symmetric CSPs (predicates whose set of accepting strings is permutation invariant). For more, see also [7], [46], [64].

c) Ordering and Ordinary CSPs: Beyond the above finite-domain CSPs, approximation resistance has been studied for *infinite-domain* CSPs (or "growing" domain CSPs). Several prominent such examples that were shown approximation resistant include Maximum Acyclic Subgraph [39], Betweenness, Non-Betweenness, Cyclic Ordering [25] and in fact, any other *ordering* CSP (CSPs over permutations of n elements) is approximation resistant [36]. Each predicate or payoff function of an ordering CSP depends on the ordering of variables on a line.

In this paper, we will use not only phylogenetic CSPs but also CSPs with finite alphabet and ordering CSPs. To distinguish finite alphabet CSPs from other CSPs, we will refer to them as *ordinary CSPs*.

II. TECHNICAL CONTRIBUTIONS AND CHALLENGES

Most closely related to our paper, both at a conceptual and technical level is the paper by Guruswami, Håstad,

[2] A technical comment is that our definition of Phylogenetic CSPs is slightly more general than the one in the logic community [17]: they only focused on unordered, rooted trees, whereas our results hold even for CSPs on *ordered* trees (left and right children are distinguishable) and on unrooted trees. Ordered trees play an important role when mapping a hierarchy to a permutation on its leaves with specific structure [14], [35], [52] and in consensus methods [51], [53].

Manokaran, Raghavendra, and Charikar (2011) [36] who showed that every ordering CSP is approximation resistant assuming the Unique Games Conjecture (see also [25], [39]). Our main technical contribution is a hardness preserving reduction from ordering CSPs to phylogenetic CSPs. At a high-level, we must deal with three main challenges:

Trees Have Topology: In Phylogenetic CSPs, whether a phylogenetic constraint is satisfied or not crucially depends on the topology of the tree. Contrast this with what happens in ordering CSPs, where simply knowing the position of an item in the permutation determines if the constraint is satisfied. For trees, the notion of "position" is more complicated and how we split the n items at internal tree nodes is important (see also the discussion on random assignments).

Many Types of Trees: Theorem I.4 provides hardness for large collections of problems studied e.g., in logic, algebra and computational biology, where trees may be ordered (left and right children are distinguishable). Contrast this with ordering CSPs where such considerations are irrelevant.

Biased Random Assignment: Perhaps counterintuitively, even the definition of a "random tree" for Phylogenetic CSPs requires some attention. Simply outputting a uniformly random tree on n leaves can result in very poor solutions. Instead, we define a natural "biased" version of a random assignment that generalizes prior methods. We show that it achieves the best possible approximation, under UGC. Contrast this with ordering CSPs, where we simply output a random permutation of n items and this is optimal.

In this paper, we present a reduction from ordering CSPs. Let us stress that a naïve reduction from ordering CSPs to phylogenetic CSPs does not give the desired hardness results. For example, the Triplet Consistency predicate $uv|w$ can be satisfied when the vertices are ordered as (u, v, w), (v, u, w), (w, u, v), and (w, v, u). So, the best hardness for Triplet Consistency we could hope for if we used the naïve reduction would be $4/3! = 2/3$.

Our main technical and conceptual contributions are as follows:

- We define a new class of biased random assignment algorithms for phylogenetic CSPs with one and many payoff functions and prove matching hardness of approximation results.
- We show that the "gap instance" from the paper by Guruswami et al. (2011) [36] can be adapted to serve as a "gadget" in the reduction from ordering to phylogenetic CSPs. A priori, it is not clear that this gap instance can be used for phylogenetic CSPs because phylogenetic CSPs are quite different from ordering CSPs. We also modify the hardness reduction by Guruswami et al. (2011) [36] to make it compatible with our own reduction. Their original reduction "erases the tree structure" of our instances because it cyclically shifts positions of variables. This preserves the relative order of most k-tuples of variables on the line but not in the tree.
- We provide a new definition of *coarse* solutions for phylogenetic CSPs. This definition substantially differs from the definition of *coarse* solutions for ordering CSPs [26], [36]. The most important difference is that we need to assign colors to different *buckets* of vertices. Without this new ingredient, it is not possible to show that every solution to the phylogenetic CSP (with payoff function f_{phy}) can be transformed to a better *coarse* solution (for an altered payoff function f^+).
- Finally, we extend our results to phylogenetic CSPs with more than one payoff function. The best algorithm for such CSPs first finds the best possible biased assignment and then uses it to obtain a random solution.

III. Preliminaries

Trees. For ease of exposition, this discussion is focused only on *ordered, rooted, binary* trees. Our results also hold for *unordered* and *unrooted* trees since phylogenetic CSPs on rooted unordered and unrooted unordered trees are special cases of phylogenetic CSPs on ordered trees. Let $T = (V, E)$ be a rooted tree with root r. A tree T is called ordered if the child nodes of every internal node u are ordered from left to right. In an ordered tree, all leaves are also ordered from left to right as in a planar drawing of that tree. In Section IX-B, we discuss an extension of our results for higher arity trees. Let us note that we will use auxiliary higher arity trees in the proof of our hardness result even for binary trees. From now on, we simply use the word "tree" to refer to ordered, rooted trees.

For $u, v \in V$, we say that u lies below v if the path from u to the root r passes through v; in this case, we may also say that v lies above u. The Lowest Common Ancestor (LCA) of a set of vertices $S \subseteq V$ is the node u that lies above all vertices in S and has the largest distance from r (the LCA node is uniquely determined by the set S).

a) Tree Homeomorphism: We now define a *phylogenetic* payoff function. Given a tree T and k distinct leaves $u_1, \ldots, u_k$ of T, a *phylogenetic* payoff function f_{phy} returns a value (payoff) in $[0, 1]$. Loosely speaking, this payoff can only depend on the relative positions of leaves $u_1, \ldots, u_k$ and their least common ancestors in the tree. Below, we formalize the definition using the notion

of homeomorphism for labeled ordered rooted trees. Then, we examine two ways of defining *phylogenetic payoff functions* using *pattern tables* and *formulas with bracket predicates*. Pattern tables correspond to truth tables of ordinary CSPs, and formulas with bracket predicates correspond to formulas with *and*, *or*, *not* predicates for ordinary CSPs.

Consider a graph G and vertex u in G of degree 2. Let v_1 and v_2 be the neighbours of u. We call the following operation *smoothing u out*: Remove vertex u along with edges (u, v_1) and (u, v_2) from G and then add edge (v_1, v_2) to G.

Definition III.1. *Consider two ordered rooted trees A and B. Let $u_1, \ldots, u_k$ be distinct leaves in A and $v_1, \ldots, v_k$ be distinct leaves in B.*

I. We say that A with labeled leaves $u_1, \ldots, u_k$ and B with labeled leaves $v_1, \ldots, v_k$ are isomorphic if there exists an isomorphism of ordered trees A and B that maps every u_i to v_i. Note that an isomorphism g of ordered trees must preserve the order of vertices. That is, if u is to the left of v, then $g(u)$ must be to the left of $g(v)$. Also, A's root must be mapped to B's root.

II. We say that A with labeled leaves $u_1, \ldots, u_k$ and B with with labeled leaves $v_1, \ldots, v_k$ are homeomorphic if A and B can be transformed to isomorphic trees A' and B' using the following three operations: (1) removing every non-labeled leaf in A or B (i.e., any leaf other than $u_1, \ldots, u_k$ in A; and any leaf other than $v_1, \ldots, v_k$ in B); (2) smoothing out vertices of degree 2 in A or B (see above for the definition); (3) removing the root if its degree is 1 and making its only child the new root.

In the definition of homeomorphic trees, we can assume that A' and B' are *irreducible trees* i.e., trees that cannot be further minimized using operations (1), (2), and (3). Note that each tree with k labeled leaves has a unique irreducible tree because operations (1), (2), and (3) commute. Consequently, the homeomorphic relation between labeled trees is transitive.

We now can formally define *phylogenetic* payoff functions.

Definition III.2. *A function f_{phy} that takes as input a tree T and k leaves $x_1, \ldots, x_k$, and returns a value in the range $[0, 1]$, is a* phylogenetic payoff function *if it satisfies the following condition: for any two trees A and B, and any leaves $u_1, \ldots, u_k$ in A and $v_1, \ldots, v_k$ in B, if A with labels $u_1, \ldots, u_k$ and B with labels $v_1, \ldots, v_k$ are homeomorphic, then $f_{phy}(A, u_1, \ldots, u_k) = f_{phy}(B, v_1, \ldots, v_k)$. The value returned by f_{phy} is called a payoff.*

To simplify notation, we will omit the tree and write $f_{phy}(x_1, \ldots, x_k)$ instead of $f_{phy}(T, x_1, \ldots, x_k)$ when it is clear that $x_1, \ldots, x_k$ are leaves of T. In this paper, we will only deal with payoff functions whose maximum payoff equals 1. We call such functions satisfiable. Before, we proceed to the definition of *phylogenetic CSPs*, we discuss how to define phylogenetic functions using *tree patterns* and *formulas with bracket predicates*.

Tree Patterns. Intuitively, a *pattern* (or motif) is a small graph that we want to find in a larger graph. Here, we are interested in *tree patterns*.

Definition III.3. *A tree pattern P is a tree with k leaves that are labeled by k variable names $x_1, \ldots, x_k$.*

We refer the reader to Section D for examples of different tree patterns.

Definition III.4. *Consider a tree T and k leaves $u_1, \ldots, u_k$ in T. We say that leaves $u_1, \ldots, u_k$ match pattern $P(x_1, \ldots, x_k)$ in T if tree T with labeled leaves $u_1, \ldots, u_k$ and P with leaves $x_1, \ldots, x_k$ are homeomorphic.*

Every (ordinary) Boolean predicate or function can be specified using a truth table. We now define an analog of a truth table for phylogenetic trees. A pattern table for f_{phy} is a list of distinct (non-homeomorphic) patterns with k variables $x_1, \ldots, x_k$ and payoffs in $[0, 1]$ assigned to the patterns. The value of phylogenetic payoff function f_{phy} defined by a pattern table on leaves $u_1, \ldots u_k$ equals to the payoff assigned to the pattern P if $u_1, \ldots, u_k$ matches $P(x_1, \ldots, x_k)$ for some pattern P in the table; and 0 if $u_1, \ldots, u_k$ do not match any pattern in the table. In Figure 4 in Appendix D, we show patterns in the pattern table for the Triplet Consistency problem; each pattern in Figure 4 is assigned a payoff of 1.

Bracket Predicates $[a, b < c]$. We can specify patterns using "bracket predicates." Consider three leaves a, b, c of a tree T. We say that $[a < b]$ if a appears to the left of b in T. We write $[a, b < c]$ if vertices a and b lie in the left subtree of $\mathrm{LCA}(a, b, c)$ and c lies in the right subtree of $\mathrm{LCA}(a, b, c)$. Similarly, we write $[a < b, c]$ if a lies in the left subtree of $\mathrm{LCA}(a, b, c)$ and b, c lie in the right subtree. It is not hard to see that every pattern can be expressed as a conjunction of bracket predicates. We show how to represent patterns for the Triplet Consistency payoff function as a conjunction of bracket predicates in Figure 4. We prove the following Lemma III.5 in Section X.

Lemma III.5. *Every pattern can be expressed as a conjunction of bracket predicates.*

In Section X, we prove that every phylogenetic payoff function can be defined using a pattern table.

Phylogenetic CSPs. A *phylogenetic CSP* problem Γ is defined by one or several phylogenetic payoff functions

$f^{(1)}_{phy}, \ldots, f^{(A)}_{phy}$ of arity k. An instance $\mathcal{I}$ of Γ consists of a set of variables V and sets of k-hyperedges $C_{f^{(i)}_{phy}}$ – one set for each predicate $f^{(i)}_{phy}$ in Γ. Thus, $\mathcal{I} = (V, C_{f^{(1)}_{phy}}, \ldots, C_{f^{(A)}_{phy}})$. A hyperedge $(u_1, \ldots, u_k)$ in $C_{f^{(i)}_{phy}}$ represents a constraint $f^{(i)}_{phy}$ on variables $u_1, \ldots, u_k$. The weight of a hyperedge $(u_1, \ldots, u_k)$ in $C_{f^{(i)}_{phy}}$ is denoted by $w_{f^{(i)}_{phy}}(u_1, \ldots, u_k)$.

In this paper, we will focus on phylogenetic CSPs with one payoff function. However, all results we prove in this paper also hold for phylogenetic CSPs with multiple payoff functions. We will discuss such CSPs in Section IX-A. When we refer to phylogenetic CSPs with one payoff function, we will omit the index i in $f^{(i)}_{phy}$ and denote the set of payoff functions by C and weights of constraints by $\mathrm{weight}(u_1, \ldots, u_k)$.

A solution for an instance $\mathcal{I}$ of phylogenetic CSP Γ is an assignment of variables V to leaves of a binary ordered tree T (the tree T is also a part of the solution). We denote the set of all solutions by $\Phi(\mathcal{I})$. The value of a solution $\varphi \in \Phi(\mathcal{I})$, which we denote by $\mathrm{val}(\varphi, \mathcal{I})$, equals the expected value of payoff functions on a random hyperedge in $\mathcal{I}$:

$$\mathrm{val}(\varphi, \mathcal{I}) = \frac{1}{\mathrm{weight}(\mathcal{I})} \times$$
$$\times \sum_{\substack{i \in \{1,\ldots,A\} \\ (x_1,\ldots,x_k) \in C_{f^{(i)}_{phy}}}} \mathrm{weight}(u_1, \ldots, u_k) f^{(i)}_{phy}(\varphi(u_1), \ldots, \varphi(u_k)),$$

where $\mathrm{weight}(\mathcal{I})$ is the total weight of all hyperedges (constraints) in $\mathcal{I}$:

$$\mathrm{weight}(\mathcal{I}) = \sum_i \sum_{(u_1,\ldots,u_k) \in C_{f^{(i)}_{phy}}} \mathrm{weight}(u_1, \ldots, u_k).$$

We denote the maximum value of a solution $\varphi \in \Phi(\mathcal{I})$ by $\mathrm{opt}(\mathcal{I})$. When the have a single payoff function f_{phy}, we denote the average value of solution φ over all constraints in $C_{f_{phy}}$ by

$$\mathrm{Avg}_{(u_1,\ldots,u_k) \in \mathcal{I}} f_{phy}(\varphi(u_1), \ldots, \varphi(u_k)).$$

Ordering Payoff Functions. We remind the reader that an ordering payoff function f_{ord} of arity k is a function that given k distinct values $x_1, \ldots, x_k$ in $\mathbb{R}$ returns a value in $[0, 1]$. The value of f_{ord} depends only on the relative order of $x_1, \ldots, x_k$. An example of an ordering payoff function is $\mathbf{1}(x_1 < x_2)$. Another example is the betweeness predicate $\mathbf{1}(x_1 < x_2 < x_3 \text{ or } x_3 < x_2 < x_1)$. Guruswami et al. (2011) [36] showed that all Ordering CSPs (i.e, CSPs with ordering payoff functions) are approximation resistant.

IV. Biased Random Assignment and Approximation Resistance

In this section, we explore definitions of *randomized assignment*, *biased randomized assignment*, and *approximation resistance*. Recall that a randomized assignment algorithm for ordinary CSPs assigns a random value from the alphabet Σ to each variable. Similarly, a random assignment algorithm for ordering CSPs permutes all variables in a random order. An analogue of these algorithms for phylogenetic CSPs randomly partitions all variables of an instance $\mathcal{I}$ into two groups and then assigns the first group to the left subtree and the second group to the right subtree. It recursively partitions variables in the left and right subtrees till each node contains at most one variable. This algorithm works well for MaxTriplets and MaxQuartets. For these problems, it gives a $1/3$ approximation. However, it drastically fails for some other phylogenetic CSPs.

Consider the following problem, which we call "split one node to the right" (see Figure 6). The payoff function $s_{phy}(x_1, \ldots, x_k)$ returns 1 if at every node when three or more variables split, only one of those variables goes to the right subtree and the other variables go to the left subtree. The abovementioned randomized assignment algorithm satisfies a predicate $s_{phy}(x_1, \ldots, x_k)$ with probability exponentially small in k. However, a biased randomized assignment algorithm that places every vertex to the left subtree with probability $1 - \delta$ and to the right subtree with probability δ satisfies predicate s with probability close to 1 if δ is sufficiently small. In Section XI, we consider a more interesting example. In that example, a randomized assignment algorithm should split vertices into two groups with probabilities that change from one recursive call to another.

The discussion above leads us to the following definition of a biased random assignment algorithm (for biased random assignments for ordinary CSPs, see [37]). A biased random assignment algorithm is specified by an absolutely continuous probability measure ρ on the interval $[0, 1]$. We remind the reader that ρ is absolutely continuous if there exists a measurable function h such that $\rho(S) = \int_S h(t)dt$ for every measurable subset S of $[0, 1]$. The measure of the interval $[0, 1]$ equals 1 because ρ is a probability measure. We assign every node of the infinite complete binary tree a subinterval of $[0, 1]$. The root of the tree is assigned $[0, 1]$. Its left child is assigned $[0, 1/2]$ and right child is assigned $[1/2, 1]$. This assignment defines weights of all nodes – the weight $\rho(u)$ equals to the measure of the interval corresponding to u.

We now assume that the algorithm is given oracle access to ρ. The algorithm recursively partitions variables in $\mathcal{I}$. Initially, it assigns all variables to the root of the binary tree. At every step, the algorithm picks a node u of the binary tree that contains more than one

variable, creates two child nodes u_{left} and u_{right}, and randomly splits all variables in u between u_{left} and u_{right}. Namely, it assigns each x in u to u_{left} and u_{right} with probabilities $\rho(u_{left})/\rho(u)$ and $\rho(u_{right})/\rho(u)$, respectively.

Let $\alpha_\rho(f_{phy})$ be the approximation factor of the biased random assignment algorithm with measure ρ for phylogenetic payoff function f_{phy} and $\alpha_{opt}(f_{phy})$ be the best approximation factor of a biased randomized assignment algorithm for f_{phy}:

$$\alpha_{opt}(f_{phy}) = \sup_{\rho} \alpha_\rho(f_{phy}).$$

As we mentioned earlier, we now consider phylogenetic CSPs with one payoff function f_{phy}. If phylogenetic CSP Γ has several payoff functions, then the algorithm should first randomly choose the appropriate measure ρ. We discuss such CSPs in Section IX-A. If a phylogenetic CSP Γ has only one payoff function, we will write $\alpha_{opt}(\Gamma) = \alpha_{\mathrm{opt}}(f_{phy})$. We call $\alpha_{opt}(\Gamma)$ the random assignment approximation factor for Γ.

We note that every measurable function h can be approximated by a piecewise constant function h'. Function h' is a constant on each interval $S_1, \ldots, S_q$ that partition $[0, 1]$ into q parts. Moreover, we can assume that the enpoints of intervals S_i are binary rational numbers. This lets us define ρ'-biased algorithms in the following equivalent way. A ρ'-biased algorithm is defined by a constant-size tree T and a probability distribution ρ' on the leaves of T. The algorithm first assigns every variable to one of the leaves of T using the distribution ρ'. Then, it recursively partitions variables splitting them 50%/50% at every step. The running time of this algorithm is linear in n (the number of variables). We have the following theorem.

Theorem IV.1. *For every phylogenetic CSP Γ and every positive ε, there exists a linear-time biased randomized rounding algorithm that has an approximation factor of $\alpha_{opt}(\Gamma) - \varepsilon$.*

We discuss the case of CSPs with more than one payoff section in Section IX-A. Finally, we define approximation resistance for phylogenetic CSPs.

Definition IV.2. *A phylogenetic CSP Γ is approximation resistant if for every positive ε, it is NP-hard to distinguish between instances of Γ that (a) have a solution of value greater than $1 - \varepsilon$ and (b) do not have a solution of value greater than $\alpha_{opt}(\Gamma) + \varepsilon$.*

We show that all phylogenetic CSPs are approximation resistant. In particular, this means that, unless $P = NP$ (assuming UGC), for every phylogenetic CSP Γ, there is no approximation algorithm with a constant approximation factor better than $\alpha_{opt}(\Gamma) + \varepsilon$.

V. Proof Overview

In this section, we give an overview of our hardness result for phylogenetic CSPs. We first show that the Triplets Consistency problem (MaxTriplets) is approximation resistant by providing a reduction from a specially crafted ordering CSP problem to the Triplets Consistency problem. This reduction works for Triplets Consistency and some other phylogenetic problems, however, fails in the general case. We then show how to modify the construction by Guruswami, Håstad, Manokaran, Raghavendra, and Charikar (2011) [36] to make our reduction work for all phylogenetic CSPs.

Consider a phylogenetic CSP Γ_{phy}. In this section, we assume that this phylogenetic CSP has only one payoff function f_{phy} of arity k. We will discuss the case when Γ_{phy} has several payoff functions in Section IX-A. Let $\mathcal{I}$ be an instance of Γ_{phy}. Denote the value of the optimal solution for $\mathcal{I}$ by $\mathrm{opt}(\mathcal{I})$. Observe that every solution φ to $\mathcal{I}$ defines an ordering on the variables of the instance $\mathcal{I}$. In this ordering, the variables are arranged from left to right according to their position in the embedding φ in the binary tree. We denote the ordering of the variables in $\mathcal{I}$ by $\mathrm{order}(\varphi)$. Let $\Phi_\pi(\mathcal{I})$ be the set of all solutions for instance $\mathcal{I}$ in which the order of variables is π (i.e., $\mathrm{order}(\varphi) = \pi$); and let $\mathrm{opt}(\mathcal{I} \mid \pi)$ be the value of the best solution φ in $\Phi_\pi(\mathcal{I})$:

$$\mathrm{opt}(\mathcal{I} \mid \pi) = \max_{\varphi \in \Phi_\pi(\mathcal{I})} \mathrm{val}(\varphi, \mathcal{I}). \tag{1}$$

Gap instance. Following [36], we use a *gap instance* $\mathcal{I}^f_{gap}$ in our reduction. The variables of this instance are leaves of an ordered perfect k-ary tree of depth d (in this tree all internal nodes have k children; and the depth of all leaves is d). Each constraint in the instance is a payoff function f on a subset of k leaves/variables. To define the instance, we introduce a random map $L_{k,m} : \{1, \ldots, k\} \to V$, where V is the set of leaves and $m = |V|$. Random map $L_{k,m}$ works as follows: it picks a random i from set $\{0, 1, \ldots, d-1\}$ and then a random (internal) node u at level i of the tree T. Let $u_1, \ldots, u_k$ be the child nodes of u arranged from left to right. In the subtrees $T_{u_1}, \ldots, T_{u_k}$ rooted at vertices $u_1, \ldots, u_k$, we independently pick random leaves $l_1, \ldots, l_k$ and map each j in $\{1, \ldots, k\}$ to l_j i.e., we let $L_{k,m}(j) = l_j$.

Now, for every k vertices $l_1, \ldots, l_k$, we add hyperedge $(l_1, \ldots, l_k)$ to the set of constraints C. The weight of $(l_1, \ldots, l_k)$ equals $\Pr\{L_{k,m}(1) = l_1, \ldots, L_{k,m}(k) = l_k\}$. Note that we use exactly the same gap instance as [36]. In their instance, the payoff function is an ordering payoff function. We will use this instance with ordering, phylogenetic, and ordinary payoff functions. In fact, we will think of $\mathcal{I}_{gap}$ as a template for k-ary CSP instances (formally, $\mathcal{I}_{gap} = C$ is the set of hyperedges).

Then, $\mathcal{I}^{f}_{gap} = (f, C)$ is an instantiation of this template with the payoff function f. Note that we can consider gap instances $\mathcal{I}^{f}_{gap}$ with ordinary, ordering, and phylogenetic payoff functions f.

Guruswami et al. (2011) [36] showed that

I. $\mathcal{I}^{f_{ord}}_{gap}$ is a completely satisfiable instance of an ordering CSP; for every ordering payoff function f_{ord} with $f_{ord}(id) = 1$, where id is the identity permutation. That is, if $f_{ord}(id) = 1$, then $\mathrm{opt}(\mathcal{I}^{f_{ord}}_{gap}) = 1$. Note that for every satisfiable payoff function f_{ord}, we can rearrange its inputs using some permutation σ so that $f_{ord} \circ \sigma(id) = 1$.

II. The cost of any so-called *coarse* solution to this instance is at most $\alpha + \varepsilon$, where α is the expected value of the random assignment algorithm (which is unique for ordering CSPs unlike phylogenetic CSPs), and ε tends to 0 as the depth d (see above) of tree T tends to infinity.

In our proof, we will also use coarse solutions. However, we postpone the discussion of such solutions till Section VI-B, where we define coarse solutions for phylogenetic CSPs (note that coarse solutions for phylogenetic CSPs and ordering CSPs differ in several important ways). In this paper, we will use the following lemma, which is an analog of property II above (for ordinary payoff functions).

Lemma V.1. *Fix natural numbers $k \geq 1$, $q \geq 1$ and positive real number $\varepsilon > 0$. Then, there exists a natural m^* such that for every template $\mathcal{I}_{gap}$ with at least m^* leaves from the family defined above, every* ordinary *payoff function $f_\circ$ of arity k defined on alphabet Σ of size q (i.e., $f_\circ$ is a function from Σ^k to $[0,1]$), the following claim holds:*

$$\mathrm{opt}(\mathcal{I}^{f_\circ}_{gap}) \leq \max_{\rho} \mathbf{E}_{x_i \sim \rho}\big[f_\circ(x_1, \ldots, x_k)\big] + \varepsilon, \qquad (2)$$

where ρ is a probability distribution on Σ; and all $x_1, \ldots, x_k$ are drawn from ρ independently.

The lemma is similar to Theorem 11.1 from the paper by Guruswami, Håstad, Manokaran, Raghavendra, and Charikar (2011) [36]. For completeness, we provide a proof of Lemma V.1 in Section VIII. To prove Lemma V.1, we rewrite bound (2) as a bound on the KL-divergence of certain random variables. Then, we prove the new bound on the KL-divergence using the chain rule for conditional entropy. We believe that our approach is somewhat simpler than the original approach used by Guruswami et al. (2011) [36].

We also show that the gap instance $\mathcal{I}^{f_{phy}}_{gap}$ is completely satisfiable for every satisfiable *phylogenetic payoff function f_{phy}*. We present a proof of the following lemma in Section VI-A. Note that this statement is not obvious for phylogenetic CSPs and does not follow from the previously known results.

Lemma V.2. *For every phylogenetic CSP Γ with payoff function f_{phy} and gap instance $\mathcal{I}^{f_{phy}}_{gap}$ of arbitrary size, we have* $\mathrm{opt}(\mathcal{I}^{f_{phy}}_{gap}) = 1$.

The main technical tool in our reduction is Theorem V.3. This theorem shows that $\mathrm{opt}(\mathcal{I}^{f_{phy}}_{gap}|\pi) \approx \alpha_{opt}(f_{phy})$ for a uniformly random permutation π. A crucial ingredient in the proof of this theorem is a new definition of coarse solutions and colorings, which we discuss in Section VI-B.

Theorem V.3. *Consider a phylogenetic CSP Γ with a payoff function f_{phy}. For every positive $\varepsilon > 0$, there exists a sufficiently large gap instance $\mathcal{I}^{f_{phy}}_{gap}$ of phylogenetic CSP Γ such that* $\mathrm{opt}(\mathcal{I}^{f_{phy}}_{gap}) = 1$*; and for a random permutation π of variables of $\mathcal{I}^{f_{phy}}_{gap}$:*

$$\mathbf{E}_{\pi}[\mathrm{opt}(\mathcal{I}^{f_{phy}}_{gap} \mid \pi)] \leq \alpha_{opt}(f_{phy}) + \varepsilon.$$

We prove this theorem in Section VI. We now discuss how to use this theorem to construct a gap preserving reduction from a certain ordering CSP problem to Γ_{phy}. Fix $\varepsilon > 0$. Let $\mathcal{I}^{f_{phy}}_{gap}$ be a gap instance from Theorem V.3 and m be the number of variables in $\mathcal{I}^{f_{phy}}_{gap}$. Define an auxiliary ordering CSP Γ_{ord} with a payoff function o of arity m. The value of $o(\pi)$ on variables $x_1, \ldots, x_m$ equals

$$o(\pi(x_1), \ldots, \pi(x_m)) = \mathrm{opt}(\mathcal{I}^{f_{phy}}_{gap} \mid \pi),$$

where the instance $\mathcal{I}^{f_{phy}}_{gap}$ is also defined on $x_1, \ldots, x_m$. In other words, the value of payoff function o with variables $x_1, \ldots, x_m$ on permutation π equals to the best solution φ for instance $\mathcal{I}^{f_{phy}}_{gap}$ with the same set of variables $x_1, \ldots, x_m$ subject to the constraint that the variables in solution φ are ordered according to π.

Reduction. Let Γ_{phy} be the class of phylogenetic CSPs with payoff function f_{phy}, and Γ_{ord} be the class of ordering CSPs with payoff function o. We now define a reduction $h_{ord \to phy}$ from CSPs Γ_{ord} to CSPs Γ_{phy}. We take an arbitrary instance $\mathcal{I}_{ord}$ of Γ_{ord} and transform it to an instance $\mathcal{I}_{phy}$ of Γ_{phy} on the same set of variables as $\mathcal{I}_{ord}$. In instance $\mathcal{I}_{ord}$, we replace every constraint $(x_{i_1}, \cdots, x_{i_m})$ for payoff function o with a copy of the gap instance $\mathcal{I}^{f_{phy}}_{gap}$ on variables $x_{i_1}, \cdots, x_{i_m}$. That is, $\mathcal{I}_{phy}$ is the union of copies of the gap instances ("gadgets") $\mathcal{I}^{f_{phy}}_{gap}$ – one "gadget" for every constraint $(x_{i_1}, \ldots, x_{i_m})$ in $\mathcal{I}_{ord}$. We denote the obtained instance of phylogenetic CSP Γ_{phy} by $\mathcal{I}_{phy}$. We let $h_{ord \to phy}(\mathcal{I}_{ord}) = \mathcal{I}_{phy}$.

Note that for every solution φ to the phylogenetic CSP $\mathcal{I}_{phy}$, there is a corresponding solution π to the ordering CSP $\mathcal{I}_{ord}$. This solution π orders all variables in $\mathcal{I}_{ord}$ in

the same way as they are ordered by solution φ in the phylogenetic tree for $\mathcal{I}_{phy}$ i.e., $\pi = \mathrm{order}(\varphi)$. The value of each payoff function o on π is greater than or equal to the value of φ on the copy of $\mathcal{I}_{gap}^{f_{phy}}$ created for o. This is the case, because φ is a possible solution for that copy of $\mathcal{I}_{gap}^{f_{phy}}$ (since the variables in φ are ordered according to π). We have the following claim.

Claim V.4. *Consider instances $\mathcal{I}_{ord}$ to $\mathcal{I}_{phy}$ of* ordering *and phylogenetic CSPs as above. Then,*

$$\mathrm{opt}(\mathcal{I}_{ord}) \geq \mathrm{opt}(\mathcal{I}_{phy}).$$

Unfortunately, we cannot claim that $\mathrm{opt}(\mathcal{I}_{ord}) = \mathrm{opt}(\mathcal{I}_{phy})$. It is possible that $\mathrm{opt}(\mathcal{I}_{ord}) \gg \mathrm{opt}(\mathcal{I}_{phy})$. This may happen if there exists an ordering of variables π such that for every constraint $\mathbf{u} = (u_1, \ldots, u_m)$ in $\mathcal{I}_{ord}$, there exists a good *local* solution $\varphi_{\mathbf{u}} \in \Phi_\pi(\mathcal{I})$:

$$o(\varphi_{\mathbf{u}}(u_1), \ldots, \varphi_{\mathbf{u}}(u_m)) \approx 1.$$

Note that $\varphi_{\mathbf{u}}$ may depend on the constraint $\mathbf{u}$. However, there is no good *global* solution $\varphi \in \Phi_\pi(\mathcal{I})$ that works for all constraints $(u_1, \ldots, u_m)$ (on average) in $\mathcal{I}_{phy}$. That is, for every $\varphi \in \Phi_\pi(\mathcal{I})$,

$$\mathrm{val}(\varphi, \mathcal{I}_{phy}) = \operatorname*{Avg}_{(u_1,\ldots,u_m)\in\mathcal{I}_{phy}} f_{phy}(\varphi(u_1), \ldots, \varphi(u_m)) \ll 1.$$

Hardness of approximation. We now discuss how to use reduction $h_{ord\to phy}$ to show that Γ_{phy} is approximation resistant. The ordering CSP Γ_{ord} is approximation resistant as every ordering CSP ([36]). By Theorem V.3, the expected value of payoff function o on a random permutation π is at most $\alpha_{opt}(f_{phy})+\varepsilon$. Hence, assuming the Unique Games conjecture, it is NP-hard to distinguish between

A. instances of Γ_{ord} that are at most $(\alpha_{opt}(f_{phy})+\varepsilon)+\varepsilon$ satisfiable; and
B. instances of Γ_{ord} that are at least $(1-\varepsilon)$ satisfiable.

To finish the proof, we would like to show that $h_{ord\to phy}$ is a gap preserving reduction. Namely, $h_{ord\to phy}$ maps (a) every instance $\mathcal{I}_{ord}$ of value at most $\alpha_{opt}(f_{phy}) + 2\varepsilon$ to an instance $\mathcal{I}_{phy}$ of value at most $\alpha_{opt}(f_{phy}) + 2\varepsilon$; and (b) every instance $\mathcal{I}_{ord}$ of value at least $1-\varepsilon$ to an instance of Γ_{phy} of value $1 - O(\varepsilon)$. If $h_{ord\to phy}$ satisfied these properties, we would conclude that, assuming UGC, it is NP-hard to distinguish between (A) instances of Γ_{phy} that are at most $\alpha_{opt}(f_{phy}) + 2\varepsilon$ satisfiable; and (B) instances of Γ_{phy} that are at least $1 - O(\varepsilon)$ satisfiable.

Property (a) immediately follows from Claim V.4 because reduction $h_{ord\to phy}$ does not increase the value of the instance. Unfortunately, property (b) is not satisfied for many payoff functions f_{phy}. Nevertheless, in the next section, we show that property (b) holds for one particular function f^*_{phy} and, consequently, the phylogenetic CSP with that payoff function f^*_{phy} is approximation resistant. We will use this result to prove that Triplets Consistency is also approximation resistant.

In Section VII, we will deal with arbitrary phylogenetic CSPs. Specifically, we will modify the hardness reduction by [36] and obtain a reduction $h_{UG\to ord}$ from Unique Games to Γ_{ord} such that the composition of reductions

$$h_{UG\to phy} = h_{ord\to phy} \circ h_{UG\to ord}$$

is gap preserving.

A. Hardness for Triplets Consistency

In this section, we define a special payoff function f^*_{phy} of arity 3 for which the hardness reduction $h_{ord\to phy}$ (described in the previous section) maps almost satisfiable instances of Γ_{ord} to almost satisfiable instances of Γ_{phy}. Fix a small $\delta \in (0,1)$. Let $\mathrm{triplet}(u,v,w)$ be the Triplet Consistency payoff function: $\mathrm{triplet}(u,v,w) = 1$, if $\mathrm{LCA}(u,w) = \mathrm{LCA}(v,w)$ (in other words, w is separated from u and v before u and v are separated); $\mathrm{triplet}(u,v,w) = 0$, otherwise. Now let $f^*_{phy}(u,v,w) = \mathrm{triplet}(u,v,w)$ if the ordering of variables u, v, and w in the phylogenetic tree is u, v, and w; $f^*_{phy}(u,v,w) = (1-\delta)\,\mathrm{triplet}(u,v,w)$, otherwise. Observe that $f^*_{phy}(u,v,w)$ is a satisfiable payoff function i.e., its maximum value is 1. Let Γ_{phy} be the phylogenetic CSP with payoff function f^*_{phy} and Γ_{ord} be the corresponding ordering CSP.

Lemma V.5. *Reduction $h_{ord\to phy}$ maps every $(1-\varepsilon)$-satisfiable instance of Γ_{ord} to a $(1-\varepsilon/\delta)$-satisfiable instance of Γ_{phy}.*

Proof. Consider a $(1-\varepsilon)$-satisfiable instance $\mathcal{I}_{ord}$ of ordering CSP Γ_{ord} and the corresponding instance $\mathcal{I}_{phy} = h_{ord\to phy}(\mathcal{I}_{ord})$ of phylogenetic CSP Γ_{phy}. Let π be the optimal solution to $\mathcal{I}_{ord}$. Consider the "left" caterpillar binary tree T with n leaves. Tree T is a binary tree in which the right child of every internal node is a leaf. We construct T by taking a path of length n and attaching a right child to every node but last (see Figure 7 in Appendix). We now define a solution for instance $\mathcal{I}_{phy}$ that maps every variable of $\mathcal{I}_{phy}$ to a leaf of T. We number all leaves in the tree from left to right. Then, we map every variable u to the leaf number $\pi(u)$. Thus, the ordering of variables in solution φ is π.

We prove that $\mathrm{val}(\varphi, \mathcal{I}_{phy}) \geq 1 - \varepsilon/\delta$. Observe that $f^*_{phy}(\varphi(u), \varphi(v), \varphi(w)) = 1$ for a triplet (u,v,w) if and only if $\pi(u) < \pi(v) < \pi(w)$ (because φ maps all vertices to the leaves of the left caterpillar tree). So, it is sufficient to show that $\pi(u) < \pi(v) < \pi(w)$ for all but at most $1 - \varepsilon/\delta$ fraction of all constraints in $\mathcal{I}_{phy}$. In other words, we need to show

$$\operatorname*{Avg}_{(u,v,w)\in\mathcal{I}_{phy}} \mathbf{1}\big(\pi(u) < \pi(v) < \pi(w)\big) \geq 1 - \varepsilon/\delta, \quad (3)$$

where $\mathbf{1}(\pi(u) < \pi(v) < \pi(w))$ is the indicator of the event $\pi(u) < \pi(v) < \pi(w)$. Recall that for every ordering constraint $\mathbf{x} = (x_1, \ldots, x_m)$ in $\mathcal{I}_{ord}$, we created a copy $\mathcal{I}_{\mathbf{x}}$ of the gap instance $\mathcal{I}_{gap}^{f^*}$. Instance $\mathcal{I}_{phy}$ is the union of instances $\mathcal{I}_{\mathbf{x}}$ over all constraints $\mathbf{x}$ in $\mathcal{I}_{ord}$. Thus,

$$\begin{aligned}\operatorname{Avg}_{(u,v,w)\in\mathcal{I}_{phy}} \mathbf{1}\left(\pi(u) < \pi(v) < \pi(w)\right) = \\ = \operatorname*{Avg}_{\mathbf{x}\in\mathcal{I}_{ord}} \operatorname*{Avg}_{(u,v,w)\in\mathcal{I}_{\mathbf{x}}} \mathbf{1}(\pi(u) < \pi(v) < \pi(w)).\end{aligned}$$

Consider an ordering constraint $\mathbf{x} = (x_1, \ldots, x_m)$ in $\mathcal{I}_{ord}$. The value of the ordering payoff function o on $\mathbf{x}$ equals (by the definition of o):

$$\begin{aligned}o(\pi(x_1), \ldots, \pi(x_m)) = \operatorname{opt}(\mathcal{I}_{gap}^{f_{phy}} \mid \pi) = \\ = \max_{\varphi_{\mathbf{x}}\in\Phi_\pi(\mathcal{I}_{phy})} \operatorname*{Avg}_{(u,v,w)\in\mathcal{I}_{\mathbf{x}}} f_{phy}^*(\varphi_{\mathbf{x}}(u), \varphi_{\mathbf{x}}(v), \varphi_{\mathbf{x}}(w)).\end{aligned}$$

Observe that

$$\begin{aligned}f_{phy}^*(\varphi_{\mathbf{x}}(u), \varphi_{\mathbf{x}}(v), \varphi_{\mathbf{x}}(w)) \leq \\ \leq (1-\delta) + \delta \cdot \mathbf{1}\left(\pi(u) < \pi(v) < \pi(w)\right).\end{aligned}$$

This is because every $\varphi_{\mathbf{x}}$ in $\Phi_{\mathbf{x}}(\mathcal{I}_{phy})$ must order u, v, w according to permutation π. So, if $\mathbf{1}(\pi(u) < \pi(v) < \pi(w)) = 0$, then $f^*(\varphi_{\mathbf{x}}(u), \varphi_{\mathbf{x}}(v), \varphi_{\mathbf{x}}(w)) \leq 1 - \delta$. Therefore,

$$\begin{aligned}o(\pi(x_1), \ldots, \pi(x_m)) \leq \\ \leq (1-\delta) + \delta \operatorname*{Avg}_{(u,v,w)\in\mathcal{I}_{\mathbf{x}}} \mathbf{1}(\pi(u) < \pi(v) < \pi(w)).\end{aligned}$$

Since π satisfies at least $(1-\varepsilon)$ fraction of all constraints in $\mathcal{I}_{ord}$, we have

$$\begin{aligned}(1-\delta) + \delta \operatorname*{Avg}_{\mathbf{x}\in\mathcal{I}_{ord}} \operatorname*{Avg}_{(u,v,w)\in\mathcal{I}_{\mathbf{x}}} \mathbf{1}\left(\pi(u) < \pi(v) < \pi(w)\right) \geq \\ \geq 1 - \varepsilon.\end{aligned}$$

This inequality implies (3). This concludes the proof of Lemma V.5. □

By Lemma V.4 and Lemma V.5, reduction $h_{ord\to phy}$ maps (a) instances of Γ_{ord} with value at most $\alpha' + 2\varepsilon$ to instances of Γ_{phy} also with value at most $\alpha' + 2\varepsilon$; and (b) almost satisfiable instances of Γ_{ord} to almost satisfiable instances of Γ_{phy}, where α' is the value of the best biased random assignment for f_{phy}^*. Therefore, phylogenetic CSP Γ_{phy} with payoff function f_{phy}^* is approximation resistant.

We now show that the Triplets Consistency problem is also approximation resistant. First, observe that $\operatorname{triplet}(u, v, w) \geq f_{phy}^*(u, v, w)$ for all variables u, v, w. Hence, $\alpha' \leq \alpha = 1/3$, where α is the value of the best biased random assignment for the Triplets Consistency problem. Then, note that a $(1-\varepsilon)$-satisfiable instance of the problem with payoff function f_{phy}^* is also a $(1-\varepsilon)$-satisfiable instance of the problem with payoff function triplet (simply because $\operatorname{triplet}(u, v, w) \geq f_{phy}^*(u, v, w)$). Finally, every instance with value at most $\alpha + \varepsilon$ with payoff function f_{phy}^* has a value at most $(\alpha + \varepsilon)/(1 - \delta)$ with payoff function triplet, because $(1 - \delta) \operatorname{triplet}(u, v, w) \leq f_{phy}^*(u, v, w)$ for all u, v, w. This implies that the Triplets Consistency problem is approximation resistant.

a) Hardness for Quartets Consistency.: Using our hardness results for triplets, a simple reduction then proves that MAXQUARTETS is also approximation resistant (see Claim D.1 in Appendix C):

Corollary V.6. *Unrooted Quartet Reconstruction (*MAXQUARTETS*) is approximation resistant, so it is UGC-hard to beat the (trivial) random assignment algorithm that achieves a $\frac{1}{3}$-approximation.*

VI. FILLING THE GAPS

In this section, we build machinery to prove Theorem V.3. First, we show that every gap instance $\mathcal{I}_{gap}^{f_{phy}}$ of a phylogenetic CSP with payoff function f_{phy} is completely satisfiable. Then, we introduce coarse solutions for phylogenetic CSPs and prove important results about such solutions. In the end of this section, we put all parts together and prove Theorem V.3.

A. Gap Instance is Completely Satisfiable

Consider a phylogenetic CSP with a satisfiable payoff function f_{phy} of arity k. Let P be the pattern of f_{phy} with a payoff of 1. Pattern P is a tree with k leaves $l_1, \ldots, l_k$ such that $f_{phy}(l_1, \ldots, l_k) = 1$. By permuting the arguments of the payoff function f_{phy}, we may assume that the leaves $l_1, \ldots, l_k$ are ordered from left to right in P. We now show that $\mathcal{I}_{gap}^{f_{phy}}$ is a satisfiable instance of Γ_{phy}. We will need the following definition.

Definition VI.1. *Consider k leaves $l_1, \ldots, l_k$ of a full tree T of arity k. Let u be their least common ancestor and $u_1, \ldots, u_k$ be the child nodes of u ordered from left to right. We say that $l_1, \ldots, l_k$ are* cousins *if each l_i is a leaf in the subtree T_{u_i} rooted at u_i. We also define a predicate* cousins*: $\operatorname{cousins}(l_1, \ldots, l_k) = 1$, if $l_1, \ldots, l_k$ are cousins; and $\operatorname{cousins}(l_1, \ldots, l_k) = 0$, otherwise.*

Lemma VI.2. *Let f_{phy} be a satisfiable phylogenetic payoff function and P be a pattern as above. Consider an instance $\mathcal{I} = (V, C)$ of phylogenetic CSP with a payoff function f_{phy} and a mapping ψ of variables V of $\mathcal{I}$ to a k-ary tree T. Then, there exists a binary tree T' with the same set of leaves as T such that the following statement holds: If $x_1, \ldots, x_k$ are mapped to cousins in T by ψ, then $f_{phy}(\psi(x_1), \ldots, \psi(x_k)) = 1$ in T'.*

Observe that in the gap instance $\mathcal{I}_{gap}^{f_{phy}}$ all payoff functions are defined on leaves that are cousins. Hence, the

conditions of Lemma VI.2 are satisfied for the identity map ψ. Therefore, we have the following immediate corollary.

Corollary VI.3. *Every gap instance $\mathcal{I}_{gap}^{f_{phy}}$ with phylogenetic payoff function f_{phy} as above is completely satisfiable.*

Proof of Lemma VI.2. Let r be the root of pattern P and $l_1, \ldots, l_k$ be its leaves. We build a binary tree T' by replacing every node and its children in T with the pattern P. See Figure 8. Formally, we define T' as follows. For every internal vertex u of T, we create a copy of pattern P. Denote it by P^u. We identify every vertex u of T with the root of P^u. Also, we identify the i-th leaf of P^u with the i-th child of u. Now consider a payoff function f_{phy} on variables $x_1, \ldots, x_k$. Suppose that $\psi(x_1), \ldots, \psi(x_k)$ are cousins in tree T. Let u be their least common ancestor and $u_1, \ldots, u_k$ be u's child nodes. Then, each $\psi(x_i)$ lies in the subtree rooted in u_i. Let us now examine where $\psi(x_1), \ldots, \psi(x_k)$ are located in the new tree T'. Each $\psi(x_i)$ also lies in the subtree rooted at u_i. However, in T', $u_1, \ldots u_k$ are not child nodes of u but rather leaves of a copy of the pattern P. Thus, $\psi(x_1), \ldots, \psi(x_k)$ match pattern P in T'. Consequently, $f_{phy}(\psi(x_1), \ldots, \psi(x_k)) = 1$ for solution ψ on phylogenetic tree T'. □

B. Coarse Solutions, Labelling, and Coloring

In this section, we define coarse solutions for phylogenetic CSPs and discuss how to measure the value of such solutions. A coarse solution embeds the set of variables V into leaves of a binary tree T. Unlike a true solution for a phylogenetic CSP, in a coarse solution, many variables can and, in most cases, will be mapped to the same leaf. A coarse solution also assigns a color to every leaf of T. We denote the leaf assigned to variable x by $\xi(x)$ and color assigned to the leaf by $\mathrm{color}(\xi(x))$. We say that a coarse solution ξ is in class $\Xi_{\varepsilon,q,\pi}(\mathcal{I})$ (where $\varepsilon \in \mathbb{R}^+$, $q \in \mathbb{N}$, π is an ordering of V) if it satisfies the following properties:[3]

1) (coarse) tree T has at most q leaves;
2) at most $\varepsilon|V|$ distinct variables have the same color; and
3) moreover, variables mapped to a leaf l are consecutive variables in ordering π.

Note that this definition differs a lot from the definition of a coarse solution for ordering CSPs. In particular, coarse solutions for ordering CSPs do not assign colors to variables.

We now define two value functions for a coarse solution ξ. Consider an instance $\mathcal{I} = (V, C)$ of phylogenetic CSP with payoff function f_{phy} and an arbitrary constraint $(x_1, \ldots, x_k) \in C$. If all variables $x_1, \ldots, x_k$ have distinct colors in the coarse solution i.e., $\mathrm{color}(\xi(x_i)) \neq \mathrm{color}(\xi(x_j))$ for all i, j, then we let

$$\begin{aligned} f^-_{phy}(\xi(x_1), \ldots, \xi(x_k)) &= f^+_{phy}(\xi(x_1), \ldots, \xi(x_k)) \\ &= f_{phy}(\xi(x_1), \cdots, \xi(x_k)), \end{aligned}$$

here $f_{phy}(\xi(x_1), \cdots, \xi(x_k))$ is well defined because all leaves $\xi(x_1), \ldots, \xi(x_k)$ are distinct. If, however, two variables have the same color (i.e., $\mathrm{color}(\xi(x_i)) = \mathrm{color}(\xi(x_j))$ for some i, j), then we let $f^-_{phy}(\xi(x_1), \ldots, \xi(x_k)) = 0$ and $f^+_{phy}(\xi(x_1), \ldots, \xi(x_k)) = 1$. We then define

$$\mathrm{val}^-(\xi, \mathcal{I}) = \operatorname*{Avg}_{(x_1,\ldots,x_k)\in\mathcal{I}} f^-_{phy}(\xi(x_1), \ldots, \xi(x_k)) \quad (4)$$

$$\mathrm{val}^+(\xi, \mathcal{I}) = \operatorname*{Avg}_{(x_1,\ldots,x_k)\in\mathcal{I}} f^+_{phy}(\xi(x_1), \ldots, \xi(x_k)). \quad (5)$$

In both expressions above, we are averaging over all constraints $(x_1, \ldots, x_k)$ in instance $\mathcal{I}$.

We will use coarse instances and value functions val^+, val^- to prove Theorem V.3. Our plan is as follows. We first show that for every (true) solution $\varphi \in \Phi_\pi$ for instance $\mathcal{I}$, there exists a coarse solution $\xi \in \Xi_{\varepsilon,q,\pi}$ with $\mathrm{val}^+(\xi, \mathcal{I}) \geq \mathrm{val}(\varphi, \mathcal{I})$ (see Lemma VI.4). We then argue that for a random ordering π and $\xi \in \Xi_{\varepsilon,q,\pi}$, we have $\mathrm{val}^+(\xi, \mathcal{I}) - \mathrm{val}^-(\xi, \mathcal{I}) \leq \varepsilon$ with high probability (see Lemma VI.10 for the precise statement). Loosely speaking, this is the case because for a random ordering π, the expected fraction of constraints $(x_1, \ldots, x_k)$ with at least two variables having the same color is very small; but $\mathrm{val}^-(\xi, f_{phy})$ and $\mathrm{val}^+(\xi, f_{phy})$ differ only on such constraints. Finally, we use Lemma VI.6 to show that $\mathrm{val}^-(\xi, \mathcal{I}^{f_{phy}}_{gap}) \leq \alpha + \varepsilon$. The above chain of inequalities implies that

$$\begin{aligned} \mathrm{opt}(\mathcal{I}^{f_{phy}}_{gap} \mid \pi) &= \max_{\varphi\in\Phi_\pi} \mathrm{val}(\varphi, \mathcal{I}^{f_{phy}}_{gap}) \\ &\leq \max_{\xi\in\Xi_{\varepsilon,q,\pi}} \mathrm{val}^+(\xi, \mathcal{I}^{f_{phy}}_{gap}) \\ &\leq \max_{\xi\in\Xi_{\varepsilon,q,\pi}} \mathrm{val}^-(\xi, \mathcal{I}^{f_{phy}}_{gap}) + \varepsilon \\ &\leq \alpha + 2\varepsilon \end{aligned}$$

with high probability if π is a random ordering of variables in $\mathcal{I}^{f_{phy}}_{gap}$.

C. How to Transform True Solution to Better Coarse Solution?

We now show that for every true solution φ for $\mathcal{I}$, there exists a coarse solution with $\mathrm{val}^+(\xi, \mathcal{I}) \geq \mathrm{val}(\varphi, \mathcal{I})$. In this coarse solution ξ the variables are ordered in the same way as in φ.

Lemma VI.4. *Consider an instance $\mathcal{I} = (V, C)$ of a phylogenetic CSP Γ_{phy}. Let $\varepsilon > 1/|V|$. For every permutation π and every solution $\varphi \in \Phi_\pi(\mathcal{I})$ for $\mathcal{I}$, there*

[3]These conditions are slightly more complex for phylogenetic CSPs on non-binary trees.

exists a coarse solution $\xi \in \Xi_{\varepsilon,q,\pi}(\mathcal{I})$ *with* $q \leq 16/\varepsilon$ *such that*

$$\mathrm{val}^{+}(\xi, \mathcal{I}) \geq \mathrm{val}(\varphi, \mathcal{I}). \tag{6}$$

Proof. Let T be the tree used in solution φ, and Λ be the set of its leaves. Solution φ maps the set of variables V to Λ. We now define a function $\lambda : \Lambda \to \Lambda$ that maps all leaves of T to at most q distinct leaves of T. This function also assigns a color to every leaf in the image. Then, we define the coarse solution $\xi = \lambda \circ \varphi$. That is, ξ uses the true solution φ to map a variable x to a leaf l and then uses function λ to assign x one of q chosen leaves of T.

Algorithm. We describe an algorithm for finding function λ. The algorithm first assigns a label to every leaf u of T and a color to every label. Then, it maps each label to an arbitrary (e.g., the leftmost) leaf of T that has that label.

Our algorithm considers all nodes of the tree in the bottom-up order. Denote the subtree rooted at node u by T_u. For every u, the algorithm either processes subtree T_u and marks T_u as processed or skips node u. It processes u if one of the following conditions is met:

- u is the root of T; or
- both the left and right subtrees of u contain at least one already processed node; or
- the number of yet unlabelled leaves in T_u is greater than $\varepsilon|V|/2$.

Note that the second item can be rephrased as follows: u is the least common ancestor (LCA) of two already processed nodes.

To process a node u, the algorithm creates four new labels LL_u, LR_u, RL_u, RR_u and assigns the same new color to all of them. It assigns the first two labels LL_u, LR_u to leaves in the left subtree of u and the second two labels RL_u, RR_u to leaves in the right subtree of u. Consider the left subtree. If it does not contain already processed nodes, then all leaves of the tree receive label LL_u. Otherwise, there should be one processed node v such that subtree T_v contains all other processed nodes in the left subtree of T_u. This node v is the least common ancestor (LCA) of all processed nodes in the left subtree of T_u. We assign label LL_u to the leaves in the left subtree of u that are to the left of T_v and label LR_u to the leaves in the left subtree of u that are to the right of T_v. We assign labels in the right subtree in a similar way. See Figure 9 in the Appendix.

Value of the solution. We now show that for the coarse solution ξ constructed above, inequality (6) holds. Consider a constraint $(x_1, \ldots, x_k)$ in $\mathcal{I}$. We need to show that $f^{+}_{phy}(\xi(x_1), \ldots, \xi(x_k)) \geq f_{phy}(\xi(x_1), \ldots, \xi(x_k))$. If at least two variables x_i and x_j have the same color, then $f^{+}_{phy}(\xi(x_1), \ldots, \xi(x_k)) = 1$ and, therefore, $f^{+}_{phy}(\xi(x_1), \ldots, \xi(x_k)) = 1 \geq f_{phy}(\xi(x_1), \ldots, \xi(x_k))$. So from now on, we assume that $x_1, \ldots, x_k$ have distinct colors.

Recall, that payoff function f_{phy} can be specified by a list of patterns and corresponding payoffs: function $f_{phy}(y_1, \ldots, y_k)$ returns a certain value if $y_1, \ldots, y_k$ match the corresponding pattern. Each pattern P can be described either by a tree with k leaves $l_1, \ldots, l_k$ or as a conjunction of *bracket* predicates of the form $[y_a < y_b]$, $[y_a, y_b < y_c]$, $[y_a < y_b, y_c]$. See Section X and Lemma III.5 for details. In this proof, we will use the latter type of pattern descriptions.

Claim VI.5. *If* $\varphi(x_1), \ldots, \varphi(x_m)$ *match a pattern* P*, then* $\xi(x_1), \ldots, \xi(x_m)$ *match the same pattern* P*. Here,* φ *is the original solution, and* ξ *is the corresponding coarse solution.*

Proof. Consider an arbitrary predicate $[y_a, y_b < y_c]$, which is a part of P. If $\varphi(x_1), \ldots, \varphi(x_m)$match P, then the predicate $[\varphi(x_a), \varphi(x_b) < \varphi(x_c)]$ must be true. We show that $[\xi(x_a), \xi(x_b) < \xi(x_c)]$ is also true.

Examine node u in T where $\varphi(x_a), \varphi(x_b), \varphi(x_c)$ are split into two groups. This node u is the least common ancestor of $\varphi(x_a)$, $\varphi(x_b)$, $\varphi(x_c)$ in tree T. Since $[\varphi(x_a), \varphi(x_b) < \varphi(x_c)]$ is true, $\varphi(x_a)$ and $\varphi(x_b)$ must belong to the left subtree of u; and $\varphi(x_c)$ must belong to the right subtree of u. Now, there are two possibilities: u was or was not processed by the algorithm.

If u was processed by the algorithm, then $\varphi(x_a), \varphi(x_b)$ received labels in the left subtree and $\varphi(x_c)$ received labels in the right subtree. In the coarse solution, labels in the left subtree are mapped to leaves in the left subtree, and labels in the right subtree mapped to leaves in the right subtree. Hence, the predicate $[\xi(x_a), \xi(x_b) < \xi(x_c)]$ is true in the coarse solution.

Suppose now that u was not processed by the algorithm but, of course, one of its ancestors was processed. Denote the first ancestor of u which was processed by v. Assume without loss of generality that u is in the left subtree of v. When the algorithm processed v, it assigned two new labels LL_v, LR_v to some leaves in T_u. These labels have the same color. Since $\varphi(x_a)$, $\varphi(x_b)$, $\varphi(x_c)$ have distinct colors, only one of them could have received label LL_v or LR_v. Therefore, the other two leaves were assigned labels before v was processed. Suppose they were assigned labels when v' was processed. Note that v' is a descendent of v and u (if v' was on the path from v to u, then v' not v would be the first ancestor of u that was processed). If v' belonged to the right subtree of u, only $\varphi(x_c)$ would be its descendant and, consequently, neither $\varphi(x_a)$ nor $\varphi(x_b)$ would receive a label when v' was processed. Hence, v' is in the left subtree of u. Therefore, $\varphi(x_c)$ must have received label LR_v, and x_a, x_b received labels in $T_{v'}$.

Since all leaves having label LR_v are to the right of leaves in subtree $T_{v'}$, we get that $[\xi(x_a), \xi(x_b) < \xi(x_c)]$ is satisfied in the coarse solution. □

This completes the proof of bound (6). It remains to bound the number of labels and colors used by the algorithm.

The size of the coarse solution. We now show that $\xi \in \Xi_{\varepsilon,q,\pi}$ (see Section VI-B for definition of $\Xi_{\varepsilon,q,\pi}$). Consider the step of the algorithm when a node u is processed. Observe that each label LL_u, LR_u, RL_u, RR_u is assigned to consecutive leaves in $\pi = \mathrm{order}(\varphi)$. All of them have the same color and no other label has that color. The number of unlabeled leaves in the left and right subtrees of u is at most $\varepsilon|V|/2$. So, the total number of leaves that get colored at this step of the algorithm is at most $\varepsilon|V|$.

We now estimate the number of leaves in the image of ξ. This number equals the number of labels we create, which, in turn, equals the number of processed vertices multiplied by 4. Each node u is processed by the algorithm because of one the three reasons provided in the definition of the algorithm. The number of nodes u processed because T_u has more than $\varepsilon|V|/2$ unlabelled leaves is at most $|V|/(\varepsilon|V|/2) = 2/\varepsilon$. The number of nodes u with at least one processed node in both the left and right subtrees of u is at most $2/\varepsilon - 1$. Additionally, the algorithm always processes the root of T. Thus, the total number of processed nodes and, consequently, number of colors is at most $4/\varepsilon$. The number of labels is upper bounded by $16/\varepsilon$. □

D. No Good Coarse Solution for Gap Instance

In the previous two sections, we proved that the gap instance $\mathcal{I}_{gap}^{f_{phy}}$ has a solution φ of value 1 and then showed how to transform every true solution to a coarse solution ξ with $\mathrm{val}^+(\xi, \mathcal{I}) \geq \mathrm{val}(\varphi, \mathcal{I})$. Thus, we know that for every ε and q, there exists an ordering π and coarse solution $\xi \in \Xi_{\varepsilon,q,\pi}(\mathcal{I})$ with $\mathrm{val}^+(\xi, \mathcal{I}_{gap}^{f_{phy}}) = 1$. We now prove that, in conrast, $\mathrm{val}^-(\xi, \mathcal{I}_{gap}^{f_{phy}}) \leq \alpha + \varepsilon$ for every $\xi \in \Xi_{\varepsilon,q,\pi}(\mathcal{I})$ if the gap instance is sufficiently large.

Lemma VI.6. *For every positive $k, q \in \mathbb{N}$ and $\varepsilon' \in (0,1)$, there exists m^* such that the following claim holds. For every phylogenetic payoff function f_{phy} of arity k, gap instance $\mathcal{I}_{gap}^{f_{phy}} = (V, C)$ with $|V| \geq m^*$, and coarse solution $\xi \in \Xi_{\varepsilon,q,\pi}$, we have:*

$$\mathrm{val}^-(\xi, \mathcal{I}) \leq \alpha + \varepsilon', \tag{7}$$

where α is the biased random assignment threshold for payoff function f_{phy}; ε and π are arbitrary.

Proof. Consider a coarse solution ξ. It maps variables of instance $\mathcal{I}_{gap}^{f_{phy}}$ to leaves of some tree T. Since $\xi \in \Xi_{\varepsilon,q,\pi}$, tree T has at most q leaves $l_1, \ldots, l_{q'}$. We view this coarse solution as a solution to an *ordinary* CSP with alphabet $l_1, \ldots, l_{q'}$ and payoff function f_{phy}^-. This function applied to variables $x_1, \ldots, x_k$ returns $f_{phy}(x_1, \ldots, x_k)$ if all colors assigned to $x_1, \ldots, x_k$ are distinct and 0, otherwise. By Lemma V.1, the value of this solution is at most $\alpha' + \varepsilon'$, where α' is the expected value of the optimal biased random assignment for payoff function f_{phy}^-.

To complete the proof, we show that $\alpha' \leq \alpha$. Consider a biased random assignment algorithm with some probability distribution ρ on labels $l_1, \ldots, l_{q'}$. We can use this distribution to define a biased randomized algorithm for phylogenetic CSP instance $\mathcal{I}_{gap}^{f_{phy}}$. The biased assignment algorithm first randomly and independently assigns all vertices V to leaves $l_1, \ldots, l_{q'}$ with probabilities $\rho(l_1), \ldots, \rho(l_{q'})$. Then, it recursively partitions vertices assigned to each leaf each time splitting vertices between the left and right subtrees with probability $50\%/50\%$ (see Section IV for details). Note that the expected value of this randomized algorithm for phylogenetic payoff function f_{phy} is greater than or equal to the value of the *ordinary* payoff function f_{phy}^-. This is the case because both payoff functions have the same value if the colors of the leaves assigned to their arguments are distinct. However, $f_{phy}^-(x_1, \ldots, x_k) = 0$ if the colors of two or more leaves x_i and x_j are the same. This implies that

$$\mathbf{E}_{x_i \sim \rho}[f_{phy}^-(x_1, \ldots, x_k)] \leq \mathbf{E}_{x_i \sim \rho}[f_{phy}(x_1, \ldots, x_k)].$$

□

E. Coarse Solutions for Random Orderings

In this section, we bound the maximum difference $\max_{\xi \in \Xi_{\varepsilon,q,\pi}}(\mathrm{val}^+(\xi, \mathcal{I}) - \mathrm{val}^-(\xi, \mathcal{I}))$ for a random ordering π. We assume that the instance $\mathcal{I}$ of phylogenetic CSP is regular. That is, the weight of constraints that contain a variable x is the same for all $x \in V$. Note that our gap instance $\mathcal{I}_{gap}^{f_{phy}}$ satisfies this condition. In the lemma below, we will use the notion of the Gaifman graph. The Gaifman graph for instance $\mathcal{I}$ of a constraint satisfaction problem is a weighted graph on the variables V of $\mathcal{I}$. The weight of edge (x_1, x_2) equals the total weight of all constraints that depend on x_1 and x_2. Given an instance $\mathcal{I}$, we construct the Gaifman graph for $\mathcal{I}$ as follows. For every constraint $(x_1, \ldots, x_k)$, we add a clique on $x_1, \ldots, x_k$ with the weight of edges equal to the weight of constraint $x_1, \ldots, x_k$. It is easy to see that if $\mathcal{I}$ is a regular instance (see above), then its Gaifman graph is also regular.

Let $H = (V, E)$ be the weighted Gaifman graph for a regular instance $\mathcal{I}$. Consider an arbitrary coarse solution $\xi \in \Xi_{\varepsilon,q,\pi}(\mathcal{I})$. Denote the total weight of monochromatic edges in H by $\mathrm{mc}(\xi, H)$: $\mathrm{mc}(\xi, H) =$

$$\mathrm{weight}\left(\{(x,y) \in E : \mathrm{color}(\xi(x)) = \mathrm{color}(\xi(y))\}\right).$$

We show $\mathrm{mc}(\xi, H)$ is small on average for a random ordering π.

Lemma VI.7. *For every $\varepsilon \in (0,1)$ and positive $q \in \mathbb{N}$, there exists $m^* = O(q\log(q/\varepsilon)/\varepsilon^2)$ such that for every regular instance $\mathcal{I} = (V, C)$ with $|V| \geq m^*$, the following bound holds:*

$$\mathbf{E}_\pi\Big[\max_{\xi \in \Xi_{\varepsilon,q,\pi}(\mathcal{I})} \mathrm{mc}(\xi, H)\Big] \leq 3\varepsilon \cdot \mathrm{weight}(E). \tag{8}$$

Here, $H = (V, E)$ is the Gaifman graph of $\mathcal{I}$; π is a random ordering of V.

Proof. Let $m = |V|$. We rescale the weights of all edges so that the weight of edges leaving any node in H equals $2/m$. Then, the total weight of all edges in H is 1. In this proof, we will ignore the tree structure of the coarse solution. The ordering π is a one-to-one mapping of V to $\{0, \ldots, m-1\}$. Thus, the coarse solution ξ defines a coloring χ on $\{0, \ldots, m-1\}$: The color of i equals the color assigned by ξ to the preimage of i. That is,

$$\chi(i) = \mathrm{color}(\xi(\pi^{-1}(i))).$$

Note that (1) ξ assigns the same color to at most εm numbers; (2) the entire set $\{0, \ldots, m-1\}$ is partitioned in at most q groups of consecutive numbers and each group receives some color (every consecutive group corresponds to a leaf in the coarse solution; different groups may have the same color). We now rephrase the statement of the lemma as follows:

$$\mathbf{E}_\pi\Big[\max_{\chi} \mathrm{mc}(\chi \circ \pi, H)\Big] \leq 3\varepsilon, \tag{9}$$

where χ is a coloring satisfying the conditions above; and $\mathrm{mc}(\chi \circ \pi, H)$ is the fraction of monochromatic edges in H with respect to the coloring $\chi \circ \pi$.

The proof follows a standard probabilistic argument. First, we estimate the number of monochromatic edges for a fixed coloring χ and random permutation π. Specifically, we argue that for a fixed coloring, the expected weight of monochromatic edges is at most ε. Then, we use Maurey's concentration inequality for permutations to show that for a typical permutation π, the maximum number of monochromatic edges $\mathrm{mc}(\chi \circ \pi, H)$ over all colorings χ is at most 2ε. This yields the desired bound (9).

Consider a fixed coloring χ. By the definition, it assigns each color to at most εm numbers. Let us now orient all edges of H in an arbitrary way. The probability that the right endpoint of an edge is assigned the number of the same color as the left endpoint is at most ε. Thus, the expected weight of monochromatic edges is at most ε.

The total number of different colorings satisfying conditions (1), and (2) above is at most $(qm)^q$, because we can specify the leftmost number in each group in at most m^q ways; we can then assign colors to these q groups in at most q^q ways.

We will now use Maurey's concentration inequality [59] (see also Theorem 5.2.6 in the book by Vershynin (2018) [79] and Theorem 13 in lecture notes by Naor (2008) [60]) to bound the probability that for a random π the weight of monochromatic edges is greater than 2ε. To this end, define the distance between two permutations or orderings π' and π'' as the fraction of x where π' and π'' differ:

$$\mathrm{dist}(\pi', \pi'') = \frac{|\{x \in V : \pi'(x) \neq \pi''(x)\}|}{|V|}.$$

A function $f : Sym(m) \to \mathbb{R}$ is L-Lipschitz if $f(\pi') - f(\pi'') \leq L \cdot \mathrm{dist}(\pi', \pi'')$ for all permutations $\pi', \pi'' \in Sym(m)$. Here, $Sym(m)$ is the group of all permutations on m elements (symmetric group).

Theorem VI.8 (Maurey)**.** *Consider an L-Lipschitz function $f : Sym(m) \to \mathbb{R}$. Let π be a random permutation in $Sym(m)$. Then,*

$$\Pr\{|f(\pi) - \mathbf{E}[f]| \geq t\} \leq 2e^{-\frac{ct^2 m}{L^2}} \tag{10}$$

for some constant $c > 0$.

We will apply Theorem VI.8 to the function $\pi \mapsto \mathrm{mc}(\chi \circ \pi, H)$. To do so, we need the following claim.

Claim VI.9. *The function $\pi \mapsto \mathrm{mc}(\chi \circ \pi, H)$ is 2-Lipschitz.*

Proof. Consider two orderings π' and π''. We split all edges of G into two sets A and B. Set A contains edges (x, y) with $\pi'(x) = \pi''(x)$ and $\pi'(y) = \pi''(y)$. Set B contains the remaining edges. Each edge with both endpoints in A is assigned the same colors by $\chi \circ \pi'$ and $\chi \circ \pi''$. Thus, it is either monochromatic with respect to both colorings $\chi \circ \pi'$ and $\chi \circ \pi''$, or not monochromatic with respect to both colorings $\chi \circ \pi'$ and $\chi \circ \pi''$. Hence, $\mathrm{mc}(\chi \circ \pi', H) - \mathrm{mc}(\chi \circ \pi'', H) \leq \mathrm{weight}(B)$. However, each edge in B is incident on a node x with $\pi'(x) \neq \pi''(x)$. Since the number of such nodes equals $\mathrm{dist}(\pi', \pi'')\,|V|$, the total weight of edges in B is at most $2\,\mathrm{dist}(\pi', \pi'')$. Here, we use that the total weight of edges incident on any fixed vertex x is $2/|V|$. This concludes the proof of Claim VI.9. □

By Maurey's concentration inequality (10), we have

$$\Pr_\pi\Big\{\mathrm{mc}(\xi \circ \pi, H) - \varepsilon \geq \varepsilon\Big\} \leq 2e^{-c'\varepsilon^2 m}$$

for some positive constant c'. We now apply the union bound over all possible colorings χ and the the following inequality:

$$\Pr_\pi\{\max_\chi \mathrm{mc}(H, \chi \circ \pi) \geq 2\varepsilon\} \leq 2e^{-c'\varepsilon^2 m} q^q m^q.$$

If $m > C'q\log(q/\varepsilon)/\varepsilon^2$ (for sufficiently large constant C'), then the right hand side of the inequality is less than ε. Since $\mathrm{mc}(\chi \circ \pi)$ is upper bounded by the total weight of all edges, which is 1, we get the desired bound (9). □

We use Lemma VI.7 to bound $\max_{\xi\in\Xi_{\varepsilon,q,\pi}}(\mathrm{val}^+(\xi,\mathcal{I}) - \mathrm{val}^-(\xi,\mathcal{I}))$.

Lemma VI.10. *For every $\varepsilon \in (0,1)$ and positive $k, q \in \mathbb{N}$, there exists $m^* = O(qk^2\log(kq/\varepsilon)/\varepsilon^2))$ such that for every regular instance $\mathcal{I}$ with at least m^* variables, the following bound holds:*

$$\mathbf{E}_\pi\Big[\max_{\xi\in\Xi_{\varepsilon,q,\pi}}(\mathrm{val}^+(\xi,\mathcal{I}) - \mathrm{val}^-(\xi,\mathcal{I}))\Big] \le \varepsilon. \qquad (11)$$

Here, π is a random ordering of variables V of the instance $\mathcal{I}$.

Proof. Let H be the Gaifman graph for instance $\mathcal{I}$. Consider a coarse solution ξ and the induced coloring of variables V. Let us now examine the definition of functions val^+ and val^- given in Equations (4) and (5). These functions differ only on payoff functions $f(\xi(x_1),\ldots,\xi(x_k))$ with two variables having the same color (i.e., $\mathrm{color}(\xi(x_i)) = \mathrm{color}(\xi(x_j))$). Thus,

$$\mathrm{val}^+(\xi,\mathcal{I}) - \mathrm{val}^-(\xi,\mathcal{I}) \le \mathrm{mc}(\xi,H).$$

Note that the total weight of all constraints in the instance $\mathcal{I}$ is 1; and the total weight of all edges in H is $k(k-1)/2$, because for every payoff function in $\mathcal{I}$, we create a clique of size k in H. We now apply Lemma VI.7 with $\varepsilon' = 2\varepsilon/(3k(k-1))$ and get inequality (11). □

F. Proof of Theorem V.3

We now complete the proof of Theorem V.3. Consider a phylogenetic payoff function f_{phy} of arity k. We assume that the arguments of f_{phy} are rearranged so that there exists an assignment φ to the variables that satisfies f_{phy} and such that the ordering of variables in φ is $x_1,\ldots,x_m$ (i.e., $\mathrm{order}(\varphi) = id$). By Corollary VI.3, the gap instance $\mathcal{I}_{gap}^{f_{phy}}$ is completely satisfiable. Suppose that the number of leaves in $\mathcal{I}_{gap}^{f_{phy}}$ is larger than some sufficiently large m^*. By Lemma VI.4, we have

$$\begin{aligned}\mathbf{E}_\pi\Big[\mathrm{opt}(\mathcal{I}_{gap}^{f_{phy}} \mid \pi)\Big] &= \mathbf{E}_\pi\Big[\max_{\varphi\in\Phi_\pi}(\mathrm{val}(\varphi,\mathcal{I}))\Big] \\ &\le \mathbf{E}_\pi\Big[\max_{\xi\in\Xi_{\varepsilon,q,\pi}}(\mathrm{val}^+(\xi,\mathcal{I}))\Big].\end{aligned}$$

By Lemma VI.10,

$$\begin{aligned}\mathbf{E}_\pi\Big[\max_{\xi\in\Xi_{\varepsilon,q,\pi}}(\mathrm{val}^+(\xi,\mathcal{I}))\Big] \le \\ \le \mathbf{E}_\pi\Big[\max_{\xi\in\Xi_{\varepsilon,q,\pi}}(\mathrm{val}^-(\xi,\mathcal{I}))\Big] + \varepsilon.\end{aligned}$$

Finally, Lemma VI.6,

$$\mathbf{E}_\pi\Big[\max_{\xi\in\Xi_{\varepsilon,q,\pi}}(\mathrm{val}^-(\xi,\mathcal{I}))\Big] \le \alpha + \varepsilon.$$

This concludes the proof of Theorem V.3.

VII. Making the Reduction from Unique Games Work

We now examine the hardness reduction by Guruswami, Håstad, Manokaran, Raghavendra, and Charikar (2011) [36] and then modify it to make it work with our own reduction $h_{ord\to phy}$. As most other hardness reductions from the Unique Games Conjecture, the hardness reduction in [36] relies on a *dictatorship test* for the problem (see [56]–[58], [65]). A dictatorship test is a special instance of the problem, in our case ordering CSP Γ_{ord}, on variables in the grid $[M]^R$. On the one hand, this instance must have a *dictator* solution φ of value at least $1-\varepsilon$. On the other hand, every τ-pseudorandom solution for this instance must have value at most $\alpha+\varepsilon$, where α is the desired approximation hardness of the problem. We remind the reader that a dictator is a function φ defined on $\mathbf{z} \in [M]^R$ that depends only on one coordinate $\mathbf{z}_j$ of $\mathbf{z}$. That coordinate j is *the dictator*. A function φ is τ-pseudorandom if $T_{1-\varepsilon}\varphi$ does not have influential coordinates i.e., coordinates with influence greater than τ (here $T_{1-\varepsilon}$ is the noise operator that "flips" every coordinate of $\mathbf{z}$ with probability ε). Note that we can pick a constant M as we wish (it can depend on ε, which we treat as a fixed constant). However, the value of R depends on the Unique Games instance we use in the reduction and is not under our control (R equals the number of labels in the Unique Games instance).

The general recipe for creating dictatorship tests was provided by Raghavendra (2008) [65] in his influential paper on optimal approximation algorithms and approximation hardness for ordinary CSPs. His dictatorship test was adapted for ordering CSPs by Guruswami et al. (2011) [36]. Also, Guruswami et al. (2011) [36] defined τ-pseudorandom functions for ordering CSPs (see Definition 4.2 in their paper) and developed tools necessary for analyzing such functions.

We now outline the dictatorship test used by Guruswami et al. (2011) [36]. We will work with the ordering predicate o of arity m defined in Section V. Guruswami et al. (2011) [36] use a gap instance $\widetilde{\mathcal{I}}_{gap}^{o}$ with M variables. This is the same gap instance[4] as we described in the proof in Section V only of a larger size and applied to the ordering predicate o. Then, for every tuple $\mathbf{s}$ of m variables $s_1,\ldots,s_m \in [M]$ (m is the

[4]For technical reasons, they take several copies of this gap instance. However, in our modified hardness reduction, we will use this instance $\widetilde{\mathcal{I}}_{gap}^{o}$ as is.

arity of the ordering CSP), they define a random map $L_{\mathbf{s}}$ that maps $s_1, \ldots, s_m$ to another m tuple in $[N]^k$ (in their case $N = M$). This map should satisfy several important conditions we examine in a moment. We give a description of the dictatorship test in Figure 2.

Dictatorship Test from [36]:

- Pick a random constraint $(s_1, \ldots, s_m)$ from $\widetilde{\mathcal{I}}^o_{gap}$.
- Draw m vectors $\mathbf{z}_1, \ldots, \mathbf{z}_m \in [N]^R$ using R independent random functions $L^{(1)}_{\mathbf{s}}, \ldots, L^{(R)}_{\mathbf{s}}$:
$$(\mathbf{z}^{(j)}_1, \ldots, \mathbf{z}^{(j)}_m) = L^{(j)}_{\mathbf{s}}(s_1, \ldots, s_m).$$
- Apply ε-noise to each $\mathbf{z}^{(j)}_i$ i.e. with probability ε, replace it with a random value in $[N]$.
- Return constraint $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ for the payoff function o.

The instance $\mathcal{I}$ of the ordering CSP Γ_{ord} generated by the dictatorship test consists of a set of variables $V = [N]^R$ and set of constraints $C = \underbrace{V \times \cdots \times V}_{m}$. The weight of each constraint $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ equals the probability that this constraint is returned by the procedure above.

Fig. 2. Dictatorship Test

The map $L_{\mathbf{s}}$ should satisfy several conditions. First, for every j, the dictatorship solution $\varphi : \mathbf{z} \mapsto \mathbf{z}_j$ (where $\mathbf{z} \in V = [M]^R$ is a variable; $\mathbf{z}_j$ is a number in $[N]$) should have value $1 - O(\varepsilon)$. In this solution, several distinct variables $\mathbf{z}$ can be mapped to the same position i; in this case, we pick a random ordering among them, but preserve their relative order with other $\mathbf{z}$'s. Then, $L_{\mathbf{s}}$ should be η-*smooth* i.e., for all $t_1, \ldots, t_m \in [N]^R$, we have (for some $\eta > 0$)

$$\Pr\left\{L_{\mathbf{s}}(s_1, \ldots, s_m) = (t_1, \ldots, t_m)\right\} \geq \eta. \tag{12}$$

The marginal distribution of each coordinate of $L_{\mathbf{s}}$ should be uniform i.e., for every $\mathbf{s} \in [M]^m$, every $i \in \{1, \ldots, m\}$, and every $t \in [N]$,

$$\Pr\{L_{\mathbf{s}}(\mathbf{s})_i = t\} = \frac{1}{N}. \tag{13}$$

Here, $L_{\mathbf{s}}(\mathbf{s})_i$ denotes the i-th coordinate of $L_{\mathbf{s}}(\mathbf{s})$. Finally, there should exist a *global* SDP solution (the same for all functions $L_{\mathbf{s}}$) that match the first and second moments of every $L_{\mathbf{s}}$.

Unfortunately, this dictatorship test instance does not work for us as is. As we discussed earlier, we need to get a hardness reduction $h_{UG \to ord}$ from Unique Games to ordering CSP Γ_{ord}, which not only maps almost satisfiable instances of Unique Games to almost satisfiable instances of Γ_{ord}, but also satisfies the following condition: The composition of hardness reductions $h_{ord \to phy} \circ h_{UG \to ord}$ maps almost satisfiable instances of Unique Games to almost satisfiable instances of our phylogenetic CSP Γ_{phy}. To satisfy this condition, we need map $L_{\mathbf{s}}$ to have one additional property. Each dictatorship solution $\varphi : \mathbf{z} \mapsto \mathbf{z}_j$ must have value at least $1 - O(\varepsilon)$ when evaluated on the phylogenetic CSP corresponding to the dictatorship test instance (i.e., the image of the dictatorship test instance under $h_{ord \to phy}$). Note that in order to make this dictatorship function $\varphi : \mathbf{z} \mapsto \mathbf{z}_j$ a valid solution to the phylogenetic CSP, we need to define a tree whose leaves are elements of $[N]$.

The map used in the paper by Guruswami et al. (2011) [36] cyclically shifts elements in $[N]$ (in their case, $N = M$). This destroys any tree structure we can define on $[N]$. Let us illustrate this point by example. Consider the Triplet Consistency constraint $uv|w$ and binary tree of depth 2 with 4 leaves 1, 2, 3, 4. This constraint is satisfied if $u = 1$, $v = 2$, $w = 3$. However, if we shift values by one, $u = 2$, $v = 3$, $w = 4$, then the constraint is no longer satisfied.

We are going to define an alternative random function L that maps all variables in $[M]$ to some larger domain $[N]$. The elements of $[N]$ are associated with leaves of a binary tree. We then let $L_{\mathbf{s}}(s_1, \ldots, s_k) = (L(s_1), \ldots, L(s_k))$ and plug these functions $L_{\mathbf{s}}$ into the dictatorship test described above.

To make the proof of [36] work for this new function L, we need to ensure that maps $L_{\mathbf{s}}$ satisfy the required conditions. Finding a global SDP solution for L is easy: We get it for free, because L is a global distribution and, as such, is a convex combination of integral solutions (each realization of L is an integral solution; it maps variables in $[M]$ to leaves in $[N]$). The smoothness condition (12) can be easily obtained by perturbing L.

Now, we show that there exists a random function $L : [M] \to [N]$ that satisfies the following conditions:

- for all $u \in [M]$ and $v \in [N]$, $\Pr\{L(u) = v\} = 1/N$ (cf. Equation (13));
- for every j and assignment $\varphi : \mathbf{z} \mapsto \mathbf{z}_j$, we have $\mathrm{val}(\varphi, h_{ord \to phy}(\mathcal{I}_{test})) \geq 1 - O(\varepsilon)$, where $\mathcal{I}_{test}$ is the dictatorship test instance obtained using function L.

Let us examine the second condition. Denote $\mathcal{I}_{red} = h_{ord \to phy}(\mathcal{I}_{test})$. Recall that instance $\mathcal{I}_{red}$ is obtained from the dictatorship test instance $\mathcal{I}_{test}$ by replacing every constraint $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ for the payoff function o with a copy of the phylogenetic gap instance $\mathcal{I}^{f_{phy}}_{gap}$ on the same set of variables $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$. We remind the reader that m is the number of leaves in the gap instance $\mathcal{I}^{f_{phy}}_{gap}$, and hence is a power of k. Similarly, M is the

number of leaves in the gap instance $\widetilde{\mathcal{I}}^o_{gap}$ and is a power of m, and, consequently, a power of k. We let $N = M^d$ for some constant d. Thus, m, M, and N are powers of k. We will associate sets $[m]$, $[M]$, and $[N]$ with leaves of k-ary trees of appropriate depths. We will also map set $[N]$ to leaves of a binary tree using Lemma VI.2. We will use this mapping to define $\mathrm{val}(\varphi, \mathcal{I}_{red})$. We now show how to construct the desired function L for a sufficiently large number N. Later, in Lemma VII.4, we will prove that L satisfies the required conditions.

Lemma VII.1. *Fix a natural $k > 1$ and consider a perfect k-ary tree T_M with M leaves labeled $0, \dots, M-1$. For every positive ε, there exists an integer N and a random map L from leaves of T_M to leaves of another k-ary tree T_N with N leaves labeled $0, \dots, N-1$ such that*

- *for every $u \in [M]$ and $v \in [N]$, we have*
$$\Pr\{L(u) = v\} = 1/N;$$
- *for every k cousins $u_1, \dots, u_k$ in T_M,*
$$\Pr\left\{ \text{cousins}(L(u_1), \cdots, L(u_k)) \right\} \geq 1 - O(\varepsilon);$$
$$\Pr\{L(u_i) = v_i \ \forall i\} > 0.$$

Remark: We define the notion of *cousins* in Section VI-A. Leaves $u_1, \dots, u_k$ are cousins in a tree of arity k if each u_i lies in the subtree rooted at the i-th child of $\mathrm{LCA}(u_1, \dots, u_k)$.

Proof. Let $N = M^d$ for $d = \frac{3M}{\varepsilon^2} \ln(\frac{M}{\varepsilon})$. We create k-ary tree T_N with N leaves $0, \dots, N-1$. We also define a set of "shortcut" edges for T_N. These edges go from level 0 to d', d' to $2d'$ and so on, where $d' = \log_k M$ is the depth of tree T_M. We will denote the tree with shortcut edges by T_{sc}. This tree has arity M.

Consider the random map $L_{M,N}$ defined in Section V. It maps $[M]$ to $[N]$. Note that it always maps leaves that are cousins in T_M to leaves that are cousins in T_N. We define L using the following well-known lemma about the optimal coupling of random variables.

Lemma VII.2 (Coupling Lemma; see e.g. [67])**.** *Consider two probability distributions $\mathcal{P}$ and $\mathcal{Q}$ on a finite domain. Suppose random variable X has distribution $\mathcal{P}$, then there exists another random variable Y having distribution $\mathcal{Q}$ such that*

$$\Pr\{X \neq Y\} = \|\mathcal{P} - \mathcal{Q}\|_{TV},$$

where $\|\mathcal{P} - \mathcal{Q}\|_{TV}$ is the total variation distance between $\mathcal{P}$ and $\mathcal{Q}$.

The random variable Y in Lemma VII.2 can be obtained from X using the maximum matching between distributions $\mathcal{P}$ and $\mathcal{Q}$. For each $u \in [M]$, we use this lemma to find a random variable $L(u)$ uniformly distributed in $[N]$ such that

$$\Pr\{L(u) \neq L_{M,N}(u)\} = \\ = \frac{1}{2} \sum_{v \in [N]} \Big| \Pr\{L_{M,N}(u) = v\} - \frac{1}{N} \Big|. \quad (14)$$

The expression on the right hand side is the total variation distance between the distribution of $L_{M,N}$ and the uniform distribution on $[N]$. We now upper bound this distance.

Claim VII.3. *For all $j \in [M]$,*

$$\frac{1}{2} \sum_{v \in [N]} \Big| \Pr\{L_{M,N}(j) = v\} - \frac{1}{N} \Big| \leq \varepsilon.$$

Proof. Consider a leaf v in M-ary tree T_{sc}. Let $v(0), \dots, v(d) = v$ be the path from the root of the tree to v. For every $j \in [M]$, we count the number of times this path goes along the j-th branch of the tree. Namely, we let $B(v, j)$ be the number of nodes $v(i)$ such that $v(i+1)$ is the j-th child of $v(i)$.

Recall that random function $L_{M,N}$ picks a random $t \in 0, \dots, d-1$ and then selects a random node u at depth t in tree T_{sc}. If for this random t, $v(t+1)$ is the j-th child of $v(t)$, then $\Pr\{L_{M,N}(j) = v \mid t\} = M/N$ (because, in this case, $L_{M,N}(j) = v$ if two events occur: $u = v(t)$, and v is randomly chosen in the subtree rooted at $v(t+1)$). Otherwise, $\Pr\{L_{M,N}(j) = v \mid t\} = 0$. Hence,

$$\Pr\{L_{M,N}(j) = v\} = \frac{M}{N} \cdot \frac{B(v, j)}{d}.$$

We have

$$\sum_{v \in [N]} \Big| \Pr\{L_{M,N}(j) = v\} - \frac{1}{N} \Big| = \\ = \sum_{v \in [N]} \Big| \frac{M}{N} \cdot \frac{B(v,j)}{d} - \frac{1}{N} \Big| \\ = \frac{M}{d} \mathbf{E}_{v \in [N]} \Big| B(v, j) - \frac{d}{M} \Big|.$$

Consider a random leaf v of T_N. The path from the root to v is a random path. Every next vertex on this path is randomly chosen among the children of the current vertex. Thus, the probability that $v(i+1)$ is the j-th child of $v(i)$ is $1/M$ for every i. Consequently, $B_{v,j}$ is the sum of d independent Bernoulli random variables with parameter $1/M$. By the Chernoff bound,

$$\Pr\left\{ \Big| B(v, j) - \frac{d}{M} \Big| \geq \varepsilon \cdot \frac{d}{M} \right\} \leq 2e^{-\frac{\varepsilon^2 (d/M)}{3}} < \frac{\varepsilon}{M}.$$

Inequality $|B(v, j) - d/M| < d$ holds always. Thus,

$$\mathbf{E}_{v \in [N]} \Big| B(v, j) - \frac{d}{M} \Big| \leq \varepsilon \cdot \frac{d}{M} + \frac{\varepsilon}{M} \cdot d = \frac{2d}{M} \varepsilon.$$

□

We now finish proof of Lemma VII.1. Random function L satisfies the first condition of Lemma VII.1 because each $L(u)$ is a random variable with uniform distribution in $[N]$. Function $L_{M,N}$ maps every set of cousins in T_N to cousins in T_M. Thus, for all cousins $u_1, \dots u_k$ in T_M, we have

$$\begin{aligned}&\Pr\big\{\text{cousins}(L(u_1),\cdots,L(u_k))\big\}\\&\quad\geq \Pr\big\{\text{cousins}(L_{M,N}(u_1),\cdots,L_{M,N}(u_k))\big\}-\varepsilon k\\&\quad= 1-\varepsilon k.\end{aligned}$$

Here, we used that $\Pr\{L_{M,N}(u)\neq L(u)\}\leq\varepsilon$ for all $u\in[M]$. This proves the second condition of function L and completes the proof of Lemma VII.1. □

We now verify that the random function L from the previous lemma can be used in the dictatorship test. Specifically, we prove that the second item of Lemma VII.1 guarantees that $\mathrm{val}(\varphi,\mathcal{I}_{red})\geq 1-O(\varepsilon)$. After that, we smooth L and plug it into the dictatorship test. The smooth variant of L returns a completely random mapping into $[N]$ with a small probability η' and with the remaining probability $1-\eta'$, it returns L.

Lemma VII.4. *Let L be the random map $L : [M]\to[N]$ from Lemma VII.1. Consider the dictatorship test instance $\mathcal{I}_{test}$ constructed using L. Let $\mathcal{I}_{test}=h_{ord\to phy}(\mathcal{I}_{test})$ be the corresponding instance of phylogenetic CSP. Finally, let φ be a solution defined as $\varphi:\mathbf{z}\mapsto\mathbf{z}_j$. Then,*

$$\begin{aligned}&\mathrm{val}(\varphi,\mathcal{I}_{red})\geq\\&\geq\min_{\substack{a_1,\dots,a_k\in[M]\\ \text{cousins}(a_1,\dots,a_k)=1}}\Pr\big\{\text{cousins}(L(a_1),\dots,L(a_k))=1\big\}-\varepsilon k.\end{aligned}$$

Proof. Observe that the value of solution φ equals

$$\mathrm{val}(\varphi,\mathcal{I}_{red})=\mathbf{E}_{\mathbf{z}_1,\dots,\mathbf{z}_m}\mathbf{E}_{(i_1,\dots,i_k)}f_{phy}(\mathbf{z}^j_{i_1},\dots,\mathbf{z}^j_{i_k}),$$

where $(\mathbf{z}_1,\dots,\mathbf{z}_m)$ is a random constraint for the the ordering payoff function o returned by the dictatorship test; and $(i_1,\dots,i_k)$ is a random constraint in the copy of $\mathcal{I}^{f_{phy}}_{gap}$ created by the reduction $h_{ord\to phy}$ for constraint $(\mathbf{z}_1,\dots,\mathbf{z}_m)$.

The probability that one of $\mathbf{z}^j_{i_1},\dots,\mathbf{z}^j_{i_k}$ is affected by ε-noise and replaced by a random value at the third step of the dictatorship test is at most εk, since each of the values is changed with probability at most ε. If $\mathbf{z}^j_{i_1},\dots,\mathbf{z}^j_{i_k}$ are not changed at the third step, then

$$(\mathbf{z}^j_{i_1},\dots,\mathbf{z}^j_{i_k})=(L^{(j)}(s_{i_1}),\dots,L^{(j)}(s_{i_k})),$$

where $(s_1,\dots,s_m)$ is a random constraint in $\tilde{\mathcal{I}}^o_{gap}$ selected at the first step of the dictatorship test. Consequently,

$$\mathrm{val}(\varphi,\mathcal{I}_{red})\geq\mathbf{E}[f_{phy}(L^{(j)}(s_{i_1}),\dots,L^{(j)}(s_{i_k}))]-\varepsilon k,$$

where the expectation on the right hand side is taken over the random choice of $s_1,\dots s_m$, $i_1,\dots,i_k$, and random realization of $L^{(j)}$. Variables in every constraint in the gap instance $\mathcal{I}^{f_{phy}}_{gap}$ are cousins; $(s_{i_1},\dots,s_{i_k})$ is a constraint in the copy of $\mathcal{I}^{f_{phy}}_{gap}$ created for constraint $(s_1,\dots,s_m)$. Thus, $s_{i_1},\dots,s_{i_k}$ are also *cousins*. Hence,

$$\begin{aligned}&\mathrm{val}(\varphi,\mathcal{I}_{red})\geq\\&\geq\min_{\substack{a_1,\dots,a_k\in[M]\\ \text{cousins}(a_1,\dots,a_k)=1}}\mathbf{E}[f_{phy}(L^{(j)}(a_1),\dots,L^{(j)}(a_k))]-\varepsilon k.\end{aligned}$$

By Lemma VI.2, $f_{phy}(L^{(j)}(a_1),\dots,L^{(j)}(a_k))=1$, if $L^{(j)}(a_1),\dots,L^{(j)}(a_k)$ are cousins. Therefore,

$$\begin{aligned}&\mathrm{val}(\varphi,\mathcal{I}_{red})\geq\\&\geq\min_{\substack{a_1,\dots,a_k\in[M]\\ \text{cousins}(a_1,\dots,a_k)=1}}\Pr\{\text{cousins}(L^{(j)}(a_1),\dots,L^{(j)}(a_k))\}\\&\qquad\qquad-\varepsilon k.\end{aligned}$$

This concludes the proof of Lemma VII.4, because $L^{(j)}$ has the same distribution as L. □

VIII. Random Solutions for Ordinary CSPs on the Gap Instance are Almost Optimal

In this section, we prove Lemma V.1. Loosely speaking, this lemma says that every solution to an *ordinary* CSP instance $\mathcal{I}^{f_\circ}_{gap}$ has value at most $\alpha+\varepsilon$, where α is the optimal *biased* random assignment for this ordinary CSP:

$$\alpha=\max_{\rho}\mathbf{E}_{x_i\sim\rho}\big[f_\circ(x_1,\dots,x_k)\big].$$

See Section V for details.

We first examine a variant of Lemma 11.3 from the paper by Guruswami et al. (2011) [36]. Instance $\mathcal{I}^{f_\circ}_{gap}$ is defined on a perfect k-ary tree T of depth d. Define a probability distribution $\mathcal{P}$ on the internal nodes of T. To draw a random vertex from $\mathcal{P}$, we first pick a random leaf u of T (with the uniform distribution). We denote the path from the root to u by $u(0),\dots,u(d-1),u(d)=u$. Then, we pick a random t from 0 to $d-1$ and output $u(t)$. Note that $u(t)$ has exactly the same distribution as the one we used in the definition of random map $L_{k,m}$ and instance $\mathcal{I}^{f_\circ}_{gap}$ in Section V.

We now consider a solution φ for an *ordinary* CSP with payoff function $f_\circ$. Let $\mu_i(T_u)$ be the fraction of leaves in subtree T_u (rooted at u) having label i (i.e., leaves l in T_u with $\varphi(l)=i$). Then, the following lemma holds.

Lemma VIII.1 (cf. Lemma 11.3 in [36])**.**

$$\begin{aligned}&\mathbf{E}_{u,t\sim\mathcal{P}}\Big[\frac{1}{k}\sum_{y\in\text{child}(u(t))}\sum_{i=1}^{q}\big|\mu_i(T_y)-\mu_i(T_{u(t)})\big|\Big]\leq\\&\qquad\leq\sqrt{\frac{2\log_2 q}{d}}.\end{aligned}$$

A variant of this lemma was proved by Guruswami et al. (2011) [36]. Their upper bound is a little worse than ours. However, it is sufficient for our purposes, and we could have used it in out proof of Lemma V.1, since it tends to 0 as d goes to infinity.

We provide a proof of Lemma VIII.1 in Section B. We now use Lemma VIII.1 to prove Lemma V.1.

Proof of Lemma V.1. Let φ be a solution for $\mathcal{I}^{f_\circ}_{gap}$ where the depth d of the tree T (see above) is sufficiently large. Specifically, $d > (k/\varepsilon)^2 \log_2 q$. The value of φ on instance $\mathcal{I}^{f_\circ}_{gap}$ equals

$$\mathbf{E}[f_\circ(\varphi(L_{k,m}(1)),\dots,\varphi(L_{k,m}(k))]$$

because payoff functions in $\mathcal{I}^{f_\circ}_{gap}$ are defined on k-tuples of leaves $(L_{k,m}(1),\dots,L_{k,m}(k))$. Define another random map $\tilde{L}_{k,m}$. This function works as $L_{k,m}$ except after choosing a random node $u(t)$, $\tilde{L}_{k,m}$ maps all numbers $1,\dots,k$ randomly into subtree rooted at $u(t)$. For each choice of $u(t)$, the total variation distance between the conditional distributions of $L_{k,m}(j)$ and $\tilde{L}_{k,m}(j)$ equals

$$\frac{1}{2}\sum_{i=1}^{q}\big|\mu_i(T_{u_j(t)}) - \mu_i(T_{u(t)})\big|.$$

Thus, we can couple $\varphi(L_{k,m}(j))$ and $\varphi(\tilde{L}_{k,m}(j))$ in such a way that (see Lemma VII.2)

$$\Pr\big(\varphi(L_{k,m}(j)) \neq \varphi(\tilde{L}_{k,m}(j)) \mid u(t)\big) = \\ = \frac{1}{2}\sum_{i=1}^{q}\big|\mu_i(T_{u_j(t)}) - \mu_i(T_{u(t)})\big|.$$

Then,

$$\Pr\big(\exists j \text{ s.t. } \varphi(L_{k,m}(j)) \neq \varphi(\tilde{L}_{k,m}(j)) \mid u(t)\big) = \\ = \frac{1}{2}\sum_{j=1}^{k}\sum_{i=1}^{q}\big|\mu_i(T_{u_j(t)}) - \mu_i(T_{u(t)})\big|.$$

Finally,

$$\Pr\big(\exists j \text{ s.t. } \varphi(L_{k,m}(j)) \neq \varphi(\tilde{L}_{k,m}(j))\big) = \\ = \frac{1}{2}\mathbf{E}_{u,t\sim\mathcal{P}}\Big[\sum_{i=1}^{q}\sum_{j=1}^{k}\big|\mu_i(T_{u_j(t)}) - \mu_i(T_{u(t)})\big|\Big].$$

By Lemma VIII.1, the right hand side is upper bounded by $k\sqrt{\frac{\log_2 q}{2d}}$. Thus,

$$\mathbf{E}\big[f_\circ(\varphi(L_{k,m}(1)),\dots,\varphi(L_{k,m}(k))\big] \leq \\ \leq \mathbf{E}\big[f_\circ(\varphi(\widetilde{L}_{k,m}(1)),\dots,\varphi(\widetilde{L}_{k,m}(k))\big] + k\sqrt{\frac{\log_2 q}{2d}}.$$

Here we used that $f_\circ$ is upper bound by 1. Now, observe that when we use function $\widetilde{L}_{k,m}(k))$, we essentially do a biased random assignment. Namely, we first pick $u(t)$ and then randomly and independently pick labels for $x_1,\dots,x_k$ in the subtree rooted at $u(t)$. It is important that after $u(t)$ is chosen all variables $x_1,\dots,x_k$ are i.i.d. Thus, the first term is upper bounded by α. We get

$$\mathbf{E}\big[f_\circ(\varphi(L_{k,m}(1)),\dots,\varphi(L_{k,m}(k))\big] \leq \\ \leq \alpha + k\sqrt{\frac{\log_2 q}{2d}} \leq \alpha + \varepsilon.$$

This concludes the proof of Lemma V.1. □

IX. Generalizations

A. Phylogenetic CSPs with Multiple Payoff Functions

We now discuss phylogenetic CSPs with multiple payoff functions $f^{(1)}_{phy},\dots,f^{(r)}_{phy}$. We assume that they are scaled so that the maximum payoff of each $f^{(i)}_{phy}$ is 1. First, consider a special case of the problem when the total weight of constraints of every type is prescribed in advance. Namely, suppose that every instance must have μ_i weight of constraints for payoff function $f^{(i)}_{phy}$ i.e.$\mathrm{weight}(C_{f^{(i)}_{phy}}) = \mu_i$. This variant of the problem is essentially equivalent to the problem with a composite payoff function f^*_μ defined as follows:

$$f^*_\mu\big(x^{(1)}_1,\dots,x^{(1)}_k,x^{(2)}_1,\dots,x^{(2)}_k,\dots,x^{(r)}_1,\dots,x^{(r)}_k\big) = \\ = \mu_i\sum_{i=1}^{r} f^{(i)}_{phy}(x^{(i)}_1,\dots,x^{(i)}_k).$$

More precisely, the phylogenetic CSP problem with payoff function f^*_μ is a special case of the problem with functions $\{f^i_{phy}\}$ and prescribed weights μ_i. This is the case simply because f^*_μ can be expressed as the sum of functions f^i_{phy}. For every μ, we know the hardness of this problem. It is defined by the approximation threshold

$$\alpha_{opt}(f^*_\mu) = \sup_\rho \alpha_\rho(f^*_\mu) = \sup_\rho \sum_{i=1}^{r}\mu_i\,\alpha_\rho(f^{(i)}_{phy}).$$

Let $\mu^* = \mathrm{argmin}_\mu(f^*_\mu)$. Our phylogenetic problem with functions $f^{(1)}_{phy},\dots,f^{(r)}_{phy}$ is at least as hard as $f^*_{\mu^*}$. Consequently, for almost satisfiable instances of phylogenetic CSPs with payoff functions $f^{(1)}_{phy},\dots,f^{(r)}_{phy}$, it is NP-hard (assuming the Unique Games Conjecture) to find a solution of value at least $\alpha_{opt}(f_1,\dots,f_r)+\varepsilon$, where

$$\alpha_{opt}(f^{(1)}_{phy},\dots,f^{(r)}_{phy}) = \alpha_{opt}(f^*_{\mu^*}) \\ = \sup_\rho \sum_{i=1}^{r}\mu_i\,\alpha_\rho(f^{(i)}_{phy}).$$

Note that approximation $\alpha_{opt}(f^{(1)}_{phy},\dots,f^{(r)}_{phy})-\varepsilon$ can be achieved. The algorithm can first find the ratios μ_i and the corresponding distribution ρ (for example, we discretize possible values of μ and store corresponding ρ in the precomputed table). Furthermore, instead of finding

the best ρ for the current weights μ, the algorithm can pick a measure ρ at random from a list of measures. This follows from von Neumann's (1928) minimax theorem.

The reader may ask if we can use the same distribution ρ for all instances of phylogenetic CSP Γ with several payoff function . It turns out that the answer is no. Consider payoff functions *one split to the left* and *one split right to the right* (see Figure 6). For every fixed distribution ρ, we can find an instance of the problem for which the biased randomized assignment satisfies exponentially small in k fraction of all constraints. However, if we first decide to satisfy only one type of predicates – *one split to the left* and *one split right to the right* – and pick the appropriate ρ for it, then we can satisfy $1/2-\varepsilon$ fraction of all constraints.

B. Higher Arity Trees

In this paper, we proved our main hardness result for *binary* phylogenetic trees. However, the same hardness result also holds for trees of an arbitrary fixed arity $r \geq 2$. To make our proof work for r-ary trees, we need to adjust the definitions of the *coarse solution* and bracket predicates, and then slightly modify the proof of Lemma VI.4. Specifically, the coarse solution must satisfy the following conditions:

1. (coarse) tree T has at most q leaves;
2. at most $\varepsilon|V|$ distinct variables have the same color; and

$3'$. moreover, every color class is the union of at most $2r$ groups of consecutive variables in ordering π.

The bracket predicates we need to use for r-ary trees have form $[u \to a, v \to b, w \to c]$. This predicate indicates that u, v, and w must be in subtrees a, b, and c of the $\mathrm{LCA}(u, v, w)$.

Finally, the algorithm from Lemma VI.4 should use more than four labels at every step of recursion. When node u is processed, it created r groups of labels, one group for each of u's children. In turn, every group has $r(r-1)$ labels. So, the total number of labels is $r^2(r-1)$. Suppose that yet unlabeled leaf l belongs to the subtree rooted at the a-th child of u. Assume that the top processed node in that tree is v. Then, l receives label (a, b, c) where b and c are indices of subtrees of $\mathrm{LCA}(v, l)$, where v and l belong to. If there are no processed nodes in the subtree rooted at the a-th child of u, all leaves in that tree receive label $(a, 0, 0)$.

X. Tree Patterns and Bracket Predicates

In this section, we prove (1) that every phylogenetic payoff function can be defined by a list of pattern and (2) every pattern can be expressed as a conjunction of bracket predicates mentioned (Lemma III.5 in Section III).

Claim X.1. *Every phylogenetic payoff function can be defined by a list of patterns (with a payoff assigned to each pattern).*

Proof. Consider a phylogenetic function f_{phy} of arity k. Let $\mathcal{P}$ be the set of all non-isomorphic irreducible patterns with k leaves labeled by $x_1, \ldots, x_k$. This set is finite because each irreducible tree has $2k-1$ nodes (it is a full binary tree with k leaves). See Section III for the definition of homeomorphic trees and reductions. Now for every pattern P with leaves $x_1, \ldots, x_k$ in $\mathcal{P}$, we compute $f_{phy}(P, x_1, \ldots, x_k)$ (the value of f_{phy} on pattern P) and assign it to pattern P. Finally, we remove all patterns with payoff 0.

We now prove that the obtained patterns define function f_{phy}. Consider an arbitrary tree T and k leaves $u_1, \ldots, u_k$. This tree with with leaves $u_1, \ldots, u_k$ can be reduced to some irreducible pattern P^*. This pattern P^* with leaves $u_1, \ldots, u_k$ and tree T with leaves $u_1, \ldots, u_k$ are homeomorphic. Thus, $f_{phy}(T, u_1, \ldots, u_k) = f_{phy}(P^*, u_1, \ldots, u_k)$. Since P^* is irreducible, it must be in the list $\mathcal{P}$. The value we assign to P^* is $f_{phy}(P^*, u_1, \ldots, u_k)$. Hence, the function defined by the list of pattern obtained above equals f_{phy}. □

To prove Lemma III.5, we need the following claim.

Claim X.2. *Consider two irreducible non-isomorphic patterns P_1 and P_2 with k leaves each labeled by $x_1, \ldots, x_k$. Then, there exists a bracket predicate such that P_1 satisfies this predicate, but P_2 does not.*

Proof. We prove this claim by induction on k. For $k = 1$, there is only pattern, so P_1 must be isomorphic to P_2. Suppose $k \geq 2$. Consider the left and right subtrees of P_1 and P_2: P_1^{left}, P_1^{right}, P_2^{left}, and P_2^{right}. Note that each tree P_1^{left}, P_1^{right}, P_2^{left}, and P_2^{right} must be non-empty because P_1 and P_2 are irreducible. Since P_1 and P_2 are not isomorphic, one of the two pairs P_1^{left} and P_1^{right} or P_2^{left} and P_2^{right} must be non-isomorphic. Suppose without loss of generality that P_1^{left} and P_1^{right} are non-isomorphic. Then, we consider two cases.

I. If P_1^{left} contains the same set of leaves as P_2^{left} (e.g. $\{x_3, x_7, x_8\}$), then we apply the inductive hypothesis to P_1^{left} and P_2^{left} and obtain the desired bracket predicate satisfied by P_1^{left} but not P_2^{left}. It is also satisfied by P_1 but not by P_2.

II. Suppose now that P_1^{left} and P_2^{left} contain different sets of variables (e.g., P_1^{left} contains $\{x_3, x_7, x_8\}$ but P_2^{left} contains $\{x_1, x_7, x_8\}$). If P_1^{left} has a variable x_i which is not in P_2^{left}, and P_2^{left} has a variable x_j which is not in P_1^{left}, then P_1 satisfies $[x_i < x_j]$ but P_2 does not. Otherwise, the set of variables in P_1^{left} must be a proper subset of variables in P_2^{left} or vice versa. Note

that if the set of variables in P_1^{left} is a proper subset of variables in P_2^{left}, then the set of variables in P_2^{right} is a proper subset of variables in P_2^{right}. In this case, let x_a be a common variable in P_1^{left} and P_2^{left}, x_c be a common variable in P_1^{right} and P_2^{right}, x_b be common variable between P_1^{right} and P_2^{left}. We have that P_1 satisfies the predicate $[x_a < x_b, x_c]$ but P_2 does not. The case when the set of variables in P_2^{left} is a proper subset of variables in P_1^{left} is handled similarly. □

Lemma III.5. *Every pattern can be expressed as a conjunction of bracket predicates.*

Proof. Let P be a given (ordered, binary) tree pattern on k leaves. We create all bracket constraints $[x_a < x_b]$, $[x_a, x_b < x_c]$, and $[x_a < x_b, x_c]$ that are satisfied in P. We show that the conjunction of all these predicates define the pattern P.

I. If tree T with leaves $u_1, \ldots, u_k$ matches pattern P with leaves $x_1, \ldots, x_k$, then T must satisfy all generated bracket constraints because P and T are homeomorphic trees and reductions defined in Section III preserve the value of every bracket predicate.

II. We now show that if T with leaves $u_1, \ldots, u_k$ does not match pattern P with leaves $x_1, \ldots, x_k$, then there there is at least one pattern in the description of P that does not match $u_1, \ldots, u_k$ in T. We reduce T with leaves $u_1, \ldots, u_k$ to an irreducible tree P' with leaves $u_1, \ldots, u_k$. Leaves $u_1, \ldots, u_k$ in this tree or pattern P' satisfy the same set of bracket predicates as in T. By Claim X.2, there exists a bracket predicate that is satisfied in P but not in P'. The same predicate is not satisfied in T.

□

XI. Example when Uniform Random Assignment Fails

In Figure 10, we provide an example of a phylogenetic predicate of $2k$ variables. If we use a biased random assignment algorithm which assigns variables to the left and right subtrees with fixed probabilities p_{left} and p_{right}, then we will satisfy an exponentially small in k fraction of all predicates.

Instead, we should split variables with probability $50\%/50\%$ in the root r of the tree. Then, in each vertex u in the left subtree of r, we will assign variables to the left part with probability $1-\delta$ and right part with probability δ. We do the opposite in the right subtree of r. If δ is sufficiently small, then the probability that we satisfy this predicate is almost the same as the probability that we split the variables into two equal groups in the root, which equals $\binom{2k}{k}/2^k = \Omega(1/\sqrt{k})$.

XII. Conclusion

Here we studied a large class of problems that have been studied in various communities that concern how to find hierarchical representation of data, when given as input a collection of local constraints among n data points. Specifically, the input is a set of local information on k items of interest (e.g., species of animals, documents, images etc.) and the goal is to aggregate it into a global hierarchy on the whole dataset of size n that closely agrees with the local information. The most basic case is when the input contains triplet constraints that give information about the relative similarity between 3 points a, b, c; such triplet queries are especially useful in crowdsourcing, databases, metric learning, logic, and computational biology. Furthermore, there are various other objectives that have been studied depending on the types of input information that is allowed and/or the properties required of the final hierarchy. Overall, the corresponding problems form a class of constraint satisfaction problems (CSPs) over hierarchies, that are called Phylogenetic CSPs and have been formally studied in the algebraic and logic communities. We note that many of the problems over hierarchies resemble at a high-level analogous formulations of well-motivated problems in the (flat) clustering and ranking literature, e.g., Correlation Clustering, Maximum Acyclic Subgraph, Betweenness etc.

Even though Phylogenetic CSPs have been studied for more than four decades, their approximability was not well-understood. The main result in the paper is that Phylogenetic CSPs are approximation resistant, meaning that they are hard-to-approximate better than a (biased) random assignment. This generalizes previously-known results for ordering CSPs, extends the definition of approximation resistance (to also allow for non-uniform randomized assignments) and it significantly augments the list of approximation resistant predicates by pointing to a large family of hard problems.

Acknowledgment

We would like to thank the anonymous reviewers for their useful feedback. VC also thanks Mohammad Mahdian for early discussions while interning in his group at Google New York in 2019.

Appendix A
History of the Problems and Further Related Work

Representing data as a tree is useful across various domains in order to describe the fine-grained relations between items of interest, or to visualize their treelike structure (e.g., in large networks) or the evolutionary history, e.g., for different species in taxonomy, and in natural languages/manuscripts in linguistics.

The problems considered here are old problems going back to more than four decades ago, to the original work of Aho, Sagiv, Szymanski, and Ullman (1981) [2] who wanted to understand how to build a hierarchical clustering given ancestry relationships for the leaves. In their paper titled "Inferring a Tree from Lowest Common Ancestors with an Application to the Optimization of Relational Expression" the explain how this seemingly unrelated problem of aggregating triplets (triplet reconstruction) has important applications in the area of relational databases. Since then problems finding hierarchical representations on data has been been studied in various communities, as we summarize below:

- Databases, Logic and Algebra: Aho et al. (1981) [2] gave the first algorithm to aggregate triplet constraints that finds a tree that satisfies all of them, if such a tree exists. Interestingly, similar algorithmic ideas were considered by Steel (1992) [75] motivated by applications in computational biology. Generalizations of the Triplet and Quartet Reconstruction problems have been intensively studied in the Computational Logic and Algebraic communities, see for example [16]–[18] and references therein. Specifically, they study CSPs over trees called Phylogenetic CSPs, which are infinite-domain CSPs and they are interested in the complexity of related problems. Interestingly, there are dichotomy results for Phylogenetic CSPs similar to the dichotomy results observed in complexity of boolean or finite-domain CSPs [21]–[23].
- Theoretical Computer Science: After the work of Aho et al. (1981) [2], many works built on improving the runtime of their algorithm using specialized data structures or studying related questions in various settings [33], [45], [55], [62], [70]. As we mentioned in the introduction, in terms of approximability not much was known. For the maximization version the best approximation was achieved by the random tree and no progress had been made. Special instances like dense instances were studied in an early work of Jiang et al. (1998) [54], where they gave a PTAS using techniques of Arora et al. (1995) [6] on instances with $m = \Omega(n^4)$ constraints. Moreover, the work of Byrka, Guillemot, and Jansson (2010) [24] studies approximation questions for maximization and minimization variants of triplet reconstruction and the work of Brodal, Fagerberg, Mailund, Pedersen, and Sand (2013) [19] gives efficient algorithms for computing distances between trees based on how the two trees differ with respect to triplets. Other methods for constructing trees or comparing trees based on quartets have also been studied in theoretical computer science, for example see the works of Alon, Snir, and Yuster (2014) [5]; Alon, Naves, and Sudakov (2016) [4]; Snir and Rao (2008, 2012) [72], [73]; Snir and Yuster (2012) [74]. Finally, the more general CSPs over trees that we studied here with the constraints involving more than 3 or 4 items, have also been studied as "subtree/supertree" aggregation methods [30], [49], [50].
- Crowdsourcing, Metric Learning and other Machine Learning Applications: Recall, a triplet $ab|c$ indicates that "a and b are more similar to each other than to c". For example, in Figure 1, we had $\{\{\texttt{lion, tiger}\}|\{\texttt{tuna}\}\}$. In the context of finding a hierarchy over the dataset, such triplets are interpreted as "must-link-before" constraints [80], which are the analogue of the popular "must-link" and "cannot-link" constraints that are used in the clustering literature [81] (notice that in HC, all points belong in the same cluster initially, and all points are separated at the leaves, so such "must-link"/"cannot-link" constraints do not apply). Triplets are especially useful in crowdsourcing and active learning. This is because humans are notoriously bad at providing accurate numerical information, but are quick and precise at comparing items (e.g., answering questions like which pair out of $\{$lion, tiger, tuna$\}$ is most similar); consequently, triplet queries (or more generally "ordinal" interactions) have been used to query users for a variety of downstream tasks like tree reconstruction or finding non-metric embeddings (also called ordinal embeddings) [3], [11], [32], [47], [48], [69], [77], [78].
- Taxonomy and Computational Biology: The study of hierarchical clustering is fundamental in evolutionary biology and the scientific field of Taxonomy tries to uncover the Tree of Life based on the evolutionary relationships among organisms (e.g., by finding similar genetic patterns in their DNA) [71]. Once again, such relationships often take the form of triplets and quartets aggregation methods [12], [20], [34], [61], [70], [75], [76].

APPENDIX B
PROOF OF LEMMA VIII.1

In this section, we will prove Lemma VIII.1 stated in Section VIII. We will focus on one label i. To simplify notation, let us call all leaves having that label red. Let T_x be the subtree of T rooted at x. Also, let $R(T_x)$ and $\mu(T_x)$ be the number of red leaves in T_x and the fraction of red leaves in T_x, respectively (for a subtree T_x of depth d', we have $\mu(T_x) = R(T_x)/k^{d'}$). We claim that for a random vertex $u(t)$ (drawn from $\mathcal{P}$) and each of its children x, the number of red leaves in T_x is close

to $R(T_{u(t)})/k$ on average. Below, we denote the set of k child nodes of $u(t)$ by $\mathrm{child}(u(t))$.

Lemma B.1. *For a random internal node $u \sim \mathcal{D}$, we have*

$$\mathbf{E}_{u,t\sim\mathcal{P}}\Big[\frac{1}{k}\sum_{y\in\mathrm{child}(u(t))}\big|\mu(T_y)-\mu(T_{u(t)})\big|\Big] \le \\ \le \mu(T)\sqrt{\frac{2\log_2 1/\mu(T)}{d}}. \quad (15)$$

Proof. We will assume that T has at least one red leaf. Define an auxiliary probability distribution $\mathcal{Q}$ on the internal nodes of the tree. Pick a random *red* vertex v in T. Then, as before, pick an independent t in $\{0,\cdots,d-1\}$ and output $v(t)$ (where $v(0),\cdots,v(d-1),v(d)=v$ is the path from the root of the tree to v). Note that in the definition of $\mathcal{P}$, we pick u uniformly among all leaves of T but in the definition of $\mathcal{Q}$, we pick v uniformly among all *red* leaves of T. Thus, $v(t)=x$ if and only if v is a red leaf in T_x and t is the depth of x in tree T. Consequently,

$$\Pr_{v,t\sim\mathcal{Q}}\{v(t)=x\} = \frac{R(T_x)}{R(T)}\cdot\frac{1}{d} \\ = \underbrace{\frac{R(T_x)/k^{d'}}{R(T)/k^d}}_{\mu(T_x)/\mu(T)}\cdot\Big(\frac{k^{d'}}{k^d}\cdot\frac{1}{d}\Big) \\ = \frac{\mu(T_x)}{\mu(T)}\cdot\Pr_{u,t\sim\mathcal{P}}\{u(t)=x\}.$$

If $\mu(T_x)\neq 0$, then

$$\Pr_{u,t\sim\mathcal{P}}\{u(t)=x\} = \frac{\mu(T)}{\mu(T_x)}\cdot\Pr_{v,t\sim\mathcal{Q}}\{v(t)=x\}.$$

Thus,

$$\mathbf{E}_{u,t\sim\mathcal{P}}\Big[\sum_{y\in\mathrm{child}(u(t))}\big|\mu(T_y)-\mu(T_{u(t)})\big|\Big] = \\ =\mathbf{E}_{v,t\sim\mathcal{Q}}\Big[\frac{\mu(T)}{\mu(T_{v(t)})}\sum_{y\in\mathrm{child}(v(t))}\big|\mu(T_y)-\mu(T_{v(t)})\big|\Big] \\ =\mu(T)\cdot\mathbf{E}_{v,t\sim\mathcal{Q}}\Big[\sum_{y\in\mathrm{child}(v(t))}\Big|\frac{\mu(T_y)}{\mu(T_{v(t)})}-1\Big|\Big] \\ =k\,\mu(T)\cdot\mathbf{E}_{v,t\sim\mathcal{Q}}\Big[\sum_{y\in\mathrm{child}(v(t))}\Big|\frac{R(T_y)}{R(T_{v(t)})}-\frac{1}{k}\Big|\Big].$$

In the expectation above, we ignore the terms with $R(T_{v(t)})=0$ — the probability of such $v(t)$ equals 0. For an internal node x of T, define two distributions, $\mathcal{A}_x$ and $\mathcal{B}_x$, on the set of its children $\mathrm{child}(x)$. The first distribution, $\mathcal{A}_x$, is the uniform distribution on $\mathrm{child}(x)$. The second distribution, $\mathcal{B}_x$, picks a y in $\mathrm{child}(x)$ with probability proportional to the number of red leaves in T_y i.e., for $y\in\mathrm{child}(x)$,

$$\Pr_{Y\sim\mathcal{B}_x}\{Y=y\} = \frac{R(T_y)}{R(T_x)}.$$

If T_x does not have any red leaves and, consequently, $R(T_x)=0$, then we let $\mathcal{B}_x$ be the uniform distribution on $\mathrm{child}(x)$.

For $y\in\mathrm{child}(v(t))$, we have

$$\sum_{y\in\mathrm{child}(v(t))}\Big|\frac{R(T_y)}{R(T_{v(t)})}-\frac{1}{k}\Big| = \\ = \sum_{y\in\mathrm{child}(v(t))}\Big|\Pr_{Y\sim\mathcal{A}_{v(t)}}\{Y=y\}-\Pr_{Y\sim\mathcal{B}_{v(t)}}\{Y=y\}\Big| \\ = 2\delta_{TV}(\mathcal{A}_{v(t)},\mathcal{B}_{v(t)}),$$

where $\delta_{TV}(\mathcal{A}_{v(t)},\mathcal{B}_{v(t)})$ is the total variation distance between $\mathcal{A}_{v(t)}$ and $\mathcal{B}_{v(t)}$. Thus,

$$\frac{1}{k}\mathbf{E}_{u,t\sim\mathcal{P}}\Big[\sum_{y\in\mathrm{child}(u(t))}\big|\mu(T_y)-\mu(T_{u(t)})\big|\Big] \le \\ \le 2\mu(T)\,\mathbf{E}_{v,t\sim\mathcal{Q}}\Big[\delta_{TV}(\mathcal{A}_{v(t)},\mathcal{B}_{v(t)})\Big]. \quad (16)$$

We will now show that the total variation distance between $\mathcal{A}_{v(t)}$ and $\mathcal{B}_{v(t)}$ is small on average for a random node $v(t)$ with $v,t\sim Q$. This will conclude the proof of the lemma.

Lemma B.2. *As before, let $\mu(T)=R(T)/k^d$ be the fraction of red leaves in tree T. Suppose $\mu(T)>0$. Then, for a random internal node $v(t)$ having distribution $\mathcal{Q}$, we have*

$$\mathbf{E}_{v,t\sim\mathcal{Q}}\Big[\delta_{TV}(\mathcal{A}_{v(t)},\mathcal{B}_{v(t)})\Big] \le \sqrt{\frac{\log_2 1/\mu}{2d}}.$$

Proof. Let v be a random red vertex in T. Random variable v takes $R(T)$ different values with probability $1/R(T)$ each. Hence, its entropy equals

$$H(v) = \log_2 R(T) = \log_2(k^d\cdot\mu(T)) \quad (17) \\ = d\,\log_2 k - \log_2 1/\mu.$$

By the chain rule of conditional entropy, we also have

$$H(v) = \sum_{i=0}^{d-1} H(v(i+1)\mid v(i)). \quad (18)$$

Observe that the conditional distribution of $v(i+1)$ given $v(i)$ is $\mathcal{B}_{v(i)}$. Thus,

$$H(v(i+1)\mid v(i)) = \mathbf{E}_v[H(\mathcal{B}_{v(i)})].$$

From (18), we have

$$\frac{H(v)}{d} = \frac{1}{d}\sum_{i=0}^{d-1}\mathbf{E}_v[H(\mathcal{B}_{v(i)})] = \mathbf{E}[H(\mathcal{B}_{v(t)})],$$

here t is a random number in $\{0,\dots,d-1\}$ and, consequently, $v(t)$ has distribution $\mathcal{Q}$. Using (17), we get

$$\mathbf{E}_{v,t\sim\mathcal{Q}}\big[H(\mathcal{B}_{v(t)})\big] = \log_2 k - \frac{\log_2 1/\mu}{d}.$$

We now rearrange the terms and obtain the following bound:

$$\mathbf{E}_{v,t\sim\mathcal{Q}}\big[\log_2 k - H(\mathcal{B}_{v(t)})\big] = \frac{\log_2 1/\mu}{d}.$$

For a fixed v and t, the support of distribution $\mathcal{B}_{v(t)}$ contains at most k distinct elements (namely, the k children of $v(t)$ or some subset of them). Hence, $H(\mathcal{B}_{v(t)}) \le \log_2 k$. Moreover, if $H(\mathcal{B}_{v(t)}) = \log_2 k$, then $\mathcal{B}_{v(t)}$ is the uniform distribution on the set of children $\mathrm{child}(v(t))$. That is, $\mathcal{B}_{v(t)} = \mathcal{A}_{v(t)}$. Thus, we interpret the expression $\log_2 k - H(\mathcal{B}_{v(t)})$ as the distance between $\mathcal{B}_{v(t)}$ and $\mathcal{A}_{v(t)}$. In fact, it is exactly equal to the Kullback–Leibler divergence between $\mathcal{B}_{v(t)}$ and $\mathcal{A}_{v(t)}$, since

$$\begin{aligned} &D_{KL}(\mathcal{B}_{v(t)} \parallel \mathcal{A}_{v(t)}) = \\ &- \sum_{y\in\mathrm{child}(v(t))} \Pr_{Y\sim\mathcal{B}_{v(t)}}\{Y=y\}\cdot\log_2\frac{1/k}{\Pr_{Y\sim\mathcal{B}_{v(t)}}\{Y=y\}} \\ &\quad = \underbrace{-\sum_{y\in\mathrm{child}(v(t))} \Pr_{Y\sim\mathcal{B}_{v(t)}}\{Y=y\}\cdot\log_2\frac{1}{k}}_{\log_2 k} \\ &\underbrace{-\sum_{y\in\mathrm{child}(v(t))} \Pr_{Y\sim\mathcal{B}_{v(t)}}\{Y=y\}\log_2\frac{1}{\Pr_{Y\sim\mathcal{B}_{v(t)}}\{Y=y\}}}_{H(\mathcal{B}_{v(t)})}. \end{aligned}$$

Therefore,

$$\mathbf{E}_{v,t\sim\mathcal{Q}}\big[D_{KL}(\mathcal{B}_{v(t)} \parallel \mathcal{A}_{v(t)})\big] = \frac{\log_2 1/\mu}{d}.$$

By Pinsker's inequality, we have

$$\begin{aligned} &\mathbf{E}_{v,t\sim\mathcal{Q}}\big[\delta_{TV}(\mathcal{B}_{v(t)},\mathcal{A}_{v(t)})\big] = \\ &\quad = \mathbf{E}_{v,t\sim\mathcal{Q}}\left[\sqrt{\frac{D_{KL}(\mathcal{B}_{v(t)} \parallel \mathcal{A}_{v(t)})}{2}}\right] \\ &\quad \le \sqrt{\mathbf{E}_{v,t\sim\mathcal{Q}}\left[\frac{D_{KL}(\mathcal{B}_{v(t)} \parallel \mathcal{A}_{v(t)})}{2}\right]} \\ &\quad = \sqrt{\frac{\log_2 1/\mu}{2d}}. \end{aligned}$$

□

We use Lemma B.2 to bound the right hand side of (16) and obtain the inequality (15). □

Lemma VIII.1 immediately follows from Lemma B.1:

$$\mathbf{E}_{u,t\sim\mathcal{P}}\Big[\frac{1}{k}\sum_{y\in\mathrm{child}(u(t))}\sum_{i=1}^{q}\big|\mu_i(T_y)-\mu_i(T_{u(t)})\big|\Big] \le$$

$$\sum_{i=1}^{q}\mu_i(T)\sqrt{\frac{2\log_2 1/\mu_i(T)}{d}} \le \sqrt{\frac{2\log_2 q}{d}}.$$

The function $t \mapsto t\sqrt{\log_2 1/t}$ is concave and $1/q\sum\mu_i(T) = 1/q$. Thus by Jensen's inequality:

$$\frac{1}{q}\sum_{i=1}^{q}\mu_i(T)\sqrt{\log_2 1/\mu_i(T)} \le \frac{1}{q}\sqrt{\log_2 q}.$$

APPENDIX C
TRIPLETS TO QUARTETS REDUCTION

As we have shown in the main part of the paper, Triplet Reconstruction MAXTRIPLETS is hard-to-approximate better than a random assignment, which achieves a $\frac{1}{3}$-approximation. A very similar situation appears for another basic problem based on arity 4 constraints:

We will need the following simple definition:

Definition C.1 (Quartet)**.** *A quartet q, denoted $q = ab|cd$, is an unrooted, unordered, trivalent*[5] *tree (see Figure 11). tree on 4 leaves a,b,c,d (see Figure 3, 11). An unrooted, unordered, trivalent tree T (containing leaves a,b,c,d) is said to be* consistent with *q (or T* satisfies *q), if the path in T between a,b is disjoint with the path in T between c,d. Otherwise, the quartet and the tree are* inconsistent with each other *(or T* violates *q). In general, quartets can also have weights* $\mathrm{weight}(ab|cd)$.

The natural optimization problem associated with Quartet Reconstruction is MAXQUARTETS:

Definition C.2 (MAXQUARTETS Problem)**.** *Given a set X of n data points and m quartets defined on data points from X, find the unrooted, unordered, trivalent tree T that is consistent with as many quartets as possible (per the definition above).*

We note that in phylogenetics the problem above is called *Unrooted Quartet Consistency*. In general, Quartet methods also have a long history and are widely deployed in computational biology [12], [34], [76]. There are other related versions of Quartet Reconstruction (where constraints and the output need to be rooted). All of our hardness results also hold for the rooted quartet reconstruction problem.

APPENDIX D
THE REDUCTION

Here we present a simple reduction from the rooted triplets consistency problem (MAXTRIPLETS) to the popular unrooted quartets consistency problem (MAXQUARTETS) that has been extensively studied [5], [54], [72], [74]. Recall that a triplet $ab|c$ is a rooted tree with 3 leaves a,b,c and the output is a binary rooted tree,

[5]Trivalent is an unrooted tree where every node has degree 3, except the leaves that have degree 1.

whereas a quartet $ab|c\gamma$ is an unrooted tree with 4 leaves and the output is an unrooted trivalent tree (every internal node has degree 3).

Claim D.1. *There is an approximation-preserving reduction from* MAXTRIPLETS *to* MAXQUARTETS.

Proof. Given an instance of MAXTRIPLETS with m triplets $t_1, t_2, \ldots, t_m$ over a set L of n labels, we create an instance of MAXQUARTETS with m quartets $q_1, q_2, \ldots, q_m$ over a set L' of $n+1$ labels as follows:

- $L' = L \cup \{\gamma\}$, where γ is a distinguished vertex to be used in order to define quartets below.
- For every triplet $t_i = a_i b_i | c_i$ of MAXTRIPLETS, we generate a quartet $q_i = a_i b_i | c_i \gamma$. Notice that γ is present in all generated quartets, and γ always appears on the side of the "outsider" item c_i for each of the triplets $a_i, b_i | c_i$. See Figure 3.

We claim that the generated quartet instance is equivalent to the triplet instance, in the sense that any candidate solution T for triplets (binary rooted tree) can be turned into a candidate solution T' for quartets (trivalent unrooted tree) that satisfies the same number of constraints, and vice versa.

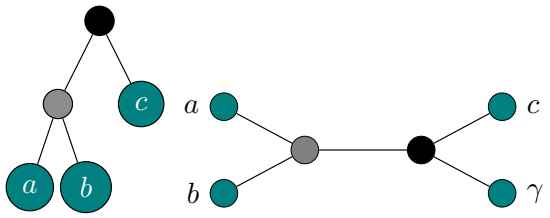

Fig. 3. The transformation of a rooted triplet $ab|c$ to an unrooted quartet $ab|c\gamma$ used in the reduction of Claim D.1.

To do so, we start with T and connect its root vertex r (that has degree 2) to another newly created vertex γ. Hence the degree of r becomes 3 and γ is a leaf (since its degree is 1). The final tree corresponds to a trivalent unrooted tree T'. Notice that a triplet $ab|c$ is satisfied by T if and only if the quartet $ab|c\gamma$ is satisfied by T', because the unique path from a to b in T is disjoint from the unique path from c to the root r and hence also to the special vertex γ. Finally, to turn any unrooted trivalent T' into a binary rooted T, we simply root T' at the special vertex γ. Then, a quartet $ab|c\gamma$ is satisfied by T' if and only if the triplet $ab|c$ is satisfied by T for the same reason as previously. □

Corollary D.2. *Unrooted Quartets Consistency (*MAXQUARTETS*) is approximation resistant, so it is UGC-hard to beat the (trivial) random assignment algorithm that achieves a $\frac{1}{3}$-approximation.*

REFERENCES

[1] Edward N Adams III. Consensus techniques and the comparison of taxonomic trees. *Systematic Biology*, 21(4):390–397, 1972.

[2] Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, and Jeffrey D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981.

[3] Noga Alon, Mihai Bădoiu, Erik D Demaine, Martin Farach-Colton, MohammadTaghi Hajiaghayi, and Anastasios Sidiropoulos. Ordinal embeddings of minimum relaxation: general properties, trees, and ultrametrics. *ACM Transactions on Algorithms (TALG)*, 4(4):1–21, 2008.

[4] Noga Alon, Humberto Naves, and Benny Sudakov. On the maximum quartet distance between phylogenetic trees. *SIAM Journal on Discrete Mathematics*, 30(2):718–735, 2016.

[5] Noga Alon, Sagi Snir, and Raphael Yuster. On the compatibility of quartet trees. *SIAM Journal on Discrete Mathematics*, 28(3):1493–1507, 2014.

[6] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 284–293, 1995.

[7] Per Austrin, Siavosh Benabbas, and Avner Magen. On quadratic threshold csps. *Discrete Mathematics & Theoretical Computer Science*, 14(Discrete Algorithms), 2012.

[8] Per Austrin and Johan Håstad. Randomly supported independence and resistance. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 483–492, 2009.

[9] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2):249–271, 2009.

[10] Dmitrii Avdiukhin, Grigory Yaroslavtsev, Danny Vainstein, Orr Fischer, Sauman Das, and Faraz Mirza. Tree learning: optimal sample complexity and algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6701–6708, 2023.

[11] Pranjal Awasthi, Maria Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. In *International Conference on Machine Learning*, pages 550–558. PMLR, 2014.

[12] Hans-Jürgen Bandelt and Andreas Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in applied mathematics*, 7(3):309–343, 1986.

[13] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56:89–113, 2004.

[14] Ziv Bar-Joseph, David K Gifford, and Tommi S Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(suppl_1):S22–S29, 2001.

[15] Amir Ben-Dor, Benny Chor, Dan Graur, Ron Ophir, and Dan Pelleg. Constructing phylogenies from quartets: elucidation of eutherian superordinal relationships. *Journal of computational Biology*, 5(3):377–390, 1998.

[16] Manuel Bodirsky. Complexity classification in infinite-domain constraint satisfaction. *arXiv preprint arXiv:1201.0856*, 2012.

[17] Manuel Bodirsky, Peter Jonsson, and Trung Van Pham. The complexity of phylogeny constraint satisfaction problems. *ACM Transactions on Computational Logic (TOCL)*, 18(3):1–42, 2017.

[18] Manuel Bodirsky and Jens K Mueller. The complexity of rooted phylogeny problems. In *Proceedings of the 13th International Conference on Database Theory*, pages 165–173, 2010.

[19] Gerth Stølting Brodal, Rolf Fagerberg, Thomas Mailund, Christian NS Pedersen, and Andreas Sand. Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1814–1832. SIAM, 2013.

[20] David Bryant. Building trees, hunting for trees, and comparing trees: theory and methods in phylogenetic analysis. *PhD Thesis*, 1997.

[21] Andrei A Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM (JACM)*, 53(1):66–120, 2006.

[22] Andrei A Bulatov. A dichotomy theorem for nonuniform csps. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330. IEEE, 2017.

[23] Andrei A Bulatov and Peter Jeavons. An algebraic approach to multi-sorted constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 183–198. Springer, 2003.

[24] Jaroslaw Byrka, Sylvain Guillemot, and Jesper Jansson. New results on optimizing rooted triplets consistency. *Discrete Applied Mathematics*, 158(11):1136–1147, 2010.

[25] Moses Charikar, Venkatesan Guruswami, and Rajsekar Manokaran. Every permutation csp of arity 3 is approximation resistant. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 62–73. IEEE, 2009.

[26] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. On the advantage over random for maximum acyclic subgraph. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 625–633. IEEE, 2007.

[27] Vaggos Chatziafratis, Mohammad Mahdian, and Sara Ahmadian. Maximizing agreements for ranking, clustering and hierarchical clustering via max-cut. In *International Conference on Artificial Intelligence and Statistics*, pages 1657–1665. PMLR, 2021.

[28] Andrew Chester, Riccardo Dondi, and Anthony Wirth. Resolving rooted triplet inconsistency by dissolving multigraphs. In *International Conference on Theory and Applications of Models of Computation*, pages 260–271. Springer, 2013.

[29] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[30] Anders Dessmark, Jesper Jansson, Andrzej Lingas, and Eva-Marta Lundell. Polynomial-time algorithms for the ordered maximum agreement subtree problem. *Algorithmica*, 48(3):233–248, 2007.

[31] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[32] Ehsan Emamjomeh-Zadeh and David Kempe. Adaptive hierarchical clustering using ordinal queries. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 415–429. SIAM, 2018.

[33] Martin Farach, Teresa M Przytycka, and Mikkel Thorup. On the agreement of many trees. *Information Processing Letters*, 55(6):297–301, 1995.

[34] Joseph Felsenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

[35] Richard F Geary, Rajeev Raman, and Venkatesh Raman. Succinct ordinal trees with level-ancestor queries. *ACM Transactions on Algorithms (TALG)*, 2(4):510–534, 2006.

[36] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the random ordering is hard: Every ordering csp is approximation resistant. *SIAM Journal on Computing*, 40(3):878–914, 2011.

[37] Venkatesan Guruswami and Euiwoong Lee. Complexity of approximating csp with balance/hard constraints. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 439–448, 2014.

[38] Venkatesan Guruswami and Euiwoong Lee. Towards a characterization of approximation resistance for symmetric csps. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[39] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 573–582. IEEE, 2008.

[40] Gustav Hast. Beating a random assignment. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 134–145. Springer, 2005.

[41] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

[42] Johan Håstad. Every 2-csp allows nontrivial approximation. In *Proceedings of the thirty-seventh Annual ACM Symposium on Theory of Computing*, pages 740–746, 2005.

[43] Johan Håstad. On the approximation resistance of a random predicate. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 149–163. Springer, 2007.

[44] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[45] Monika Rauch Henzinger, Valerie King, and Tandy Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24(1):1–13, 1999.

[46] Neng Huang and Aaron Potechin. On the approximability of presidential type predicates. *arXiv preprint arXiv:1907.04451*, 2019.

[47] Lalit Jain, Kevin G Jamieson, and Rob Nowak. Finite sample prediction and recovery bounds for ordinal embedding. *Advances in neural information processing systems*, 29, 2016.

[48] Kevin G Jamieson and Robert D Nowak. Low-dimensional embedding using adaptively selected ordinal data. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1077–1084. IEEE, 2011.

[49] Jesper Jansson, Richard S Lemence, and Andrzej Lingas. The complexity of inferring a minimally resolved phylogenetic supertree. *SIAM Journal on Computing*, 41(1):272–291, 2012.

[50] Jesper Jansson, Joseph H-K Ng, Kunihiko Sadakane, and Wing-Kin Sung. Rooted maximum agreement supertrees. *Algorithmica*, 43:293–307, 2005.

[51] Jesper Jansson, Nguyen Bao Nguyen, and Wing-Kin Sung. Algorithms for combining rooted triplets into a galled phylogenetic network. *SIAM Journal on Computing*, 35(5):1098–1121, 2006.

[52] Jesper Jansson, Kunihiko Sadakane, and Wing-Kin Sung. Ultra-succinct representation of ordered trees. In *SODA*, volume 7, pages 575–584, 2007.

[53] Jesper Jansson, Chuanqi Shen, and Wing-Kin Sung. Improved algorithms for constructing consensus trees. *Journal of the ACM (JACM)*, 63(3):1–24, 2016.

[54] Tao Jiang, Paul Kearney, and Ming Li. Orchestrating quartets: approximation and data correction. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 416–425. IEEE, 1998.

[55] Sampath Kannan, Tandy Warnow, and Shibu Yooseph. Computing the local consensus of trees. *SIAM Journal on Computing*, 27(6):1695–1724, 1998.

[56] S Khot and O Regev. Vertex cover might be hard to approximate to within 2-/spl epsiv. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 379–386. IEEE, 2003.

[57] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002.

[58] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[59] Bernard Maurey. Construction de suites symétriques. *CR Acad. Sci. Paris Sér. AB*, 288(14):A679–A681, 1979.

[60] Assaf Naor. Concentration of measure. https://web.math.princeton.edu/~naor/homepagefiles/ConcentrationofMeasure.pdf, 2008.

[61] Meei Pyng Ng, Mike Steel, and Nicholas C Wormald. The difficulty of constructing a leaf-labelled tree including or avoiding given subtrees. *Discrete Applied Mathematics*, 98(3):227–235, 2000.

[62] Meei Pyng Ng and Nicholas C Wormald. Reconstruction of rooted trees from subtrees. *Discrete applied mathematics*, 69(1-2):19–31, 1996.

[63] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[64] Aaron Potechin. On the approximation resistance of balanced linear threshold functions. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 430–441, 2019.

[65] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254, 2008.

[66] Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical review E*, 67(2):026112, 2003.

[67] Sebastien Roch. Modern discrete probability: An essential toolkit. https://people.math.wisc.edu/~roch/mdp/roch-mdp-chap4.pdf, 2022.

[68] Michael J Sanderson, Andy Purvis, and Chris Henze. Phylogenetic supertrees: assembling the trees of life. *Trends in Ecology & Evolution*, 13(3):105–109, 1998.

[69] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16, 2003.

[70] Charles Semple and Mike Steel. A supertree method for rooted trees. *Discrete Applied Mathematics*, 105(1-3):147–158, 2000.

[71] Peter HA Sneath and Robert R Sokal. *Numerical taxonomy. The principles and practice of numerical classification.* W.H. Freeman and Company; 1st edition (January 1, 1963), 1963.

[72] Sagi Snir and Satish Rao. Quartets maxcut: a divide and conquer quartets algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(4):704–718, 2008.

[73] Sagi Snir and Satish Rao. Quartet maxcut: a fast algorithm for amalgamating quartet trees. *Molecular phylogenetics and evolution*, 62(1):1–8, 2012.

[74] Sagi Snir and Raphael Yuster. Reconstructing approximate phylogenetic trees from quartet samples. *SIAM Journal on Computing*, 41(6):1466–1480, 2012.

[75] Michael Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of classification*, 9(1):91–116, 1992.

[76] Korbinian Strimmer and Arndt Von Haeseler. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. *Molecular biology and evolution*, 13(7):964–969, 1996.

[77] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. *28th International Conference on Machine Learning (ICML)*, 2011.

[78] Yoshikazu Terada and Ulrike Luxburg. Local ordinal embedding. In *International Conference on Machine Learning*, pages 847–855. PMLR, 2014.

[79] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[80] Sharad Vikram and Sanjoy Dasgupta. Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090. PMLR, 2016.

[81] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.

[82] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[83] Uri Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *SODA*, volume 98, pages 201–210, 1998.

FIGURES

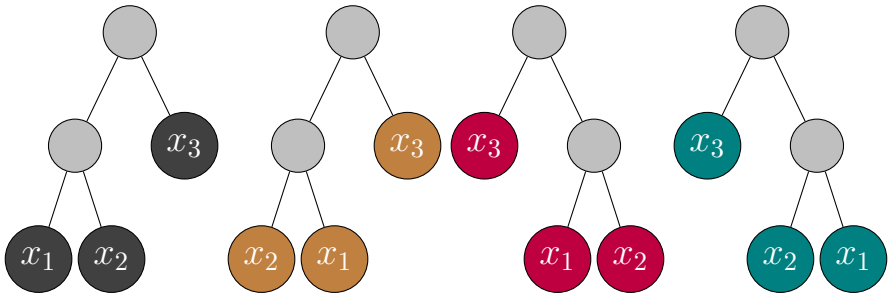

Fig. 4. Four patterns that define the Triplets Consistency problem. These pattern can also be specified using the "square brackets notation". First pattern: $[x_1, x_2 < x_3]$ & $[x_1 < x_2]$. Second pattern: $[x_1, x_2 < x_3]$ & $[x_2 < x_1]$. Third pattern: $[x_3 < x_1, x_2]$ & $[x_1 < x_2]$. Fourth pattern: $[x_3 < x_1, x_2]$ & $[x_2 < x_1]$.

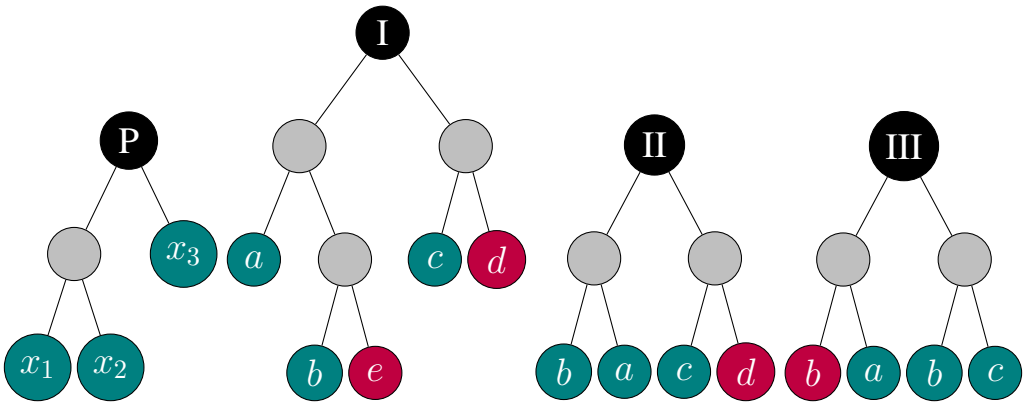

Fig. 5. Consider the leftmost tree P above. It is a pattern on variables x_1, x_2, x_3. Let f_{phy} be payoff function defined by this pattern. Namely, let $f_{phy}(a, b, c) = 1$, if a, b, c match P; 0, otherwise. Then, $f_{phy}(a, b, c) = 1$ for the tree I. However, $f_{phy}(a, b, c) = 0$ for tree II, because a and b are ordered incorrectly. Also, $f_{phy}(a, b, c) = 0$ for tree III, because a is the first node that splits from a,b, and c.

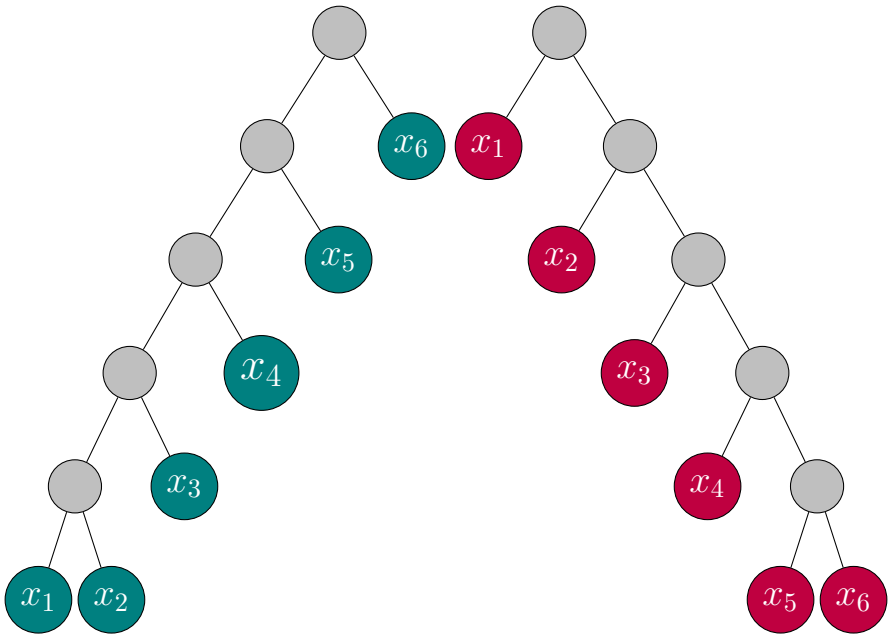

Fig. 6. The left tree is a pattern for the *split-one-to-the-right constraint*. The right tree is a pattern for the *split-one-to-the-left constraint*. Each of the constraints contains all 6! permutations of variables $x_1, \ldots, x_6$. So, the order in which variables split from others is not important.

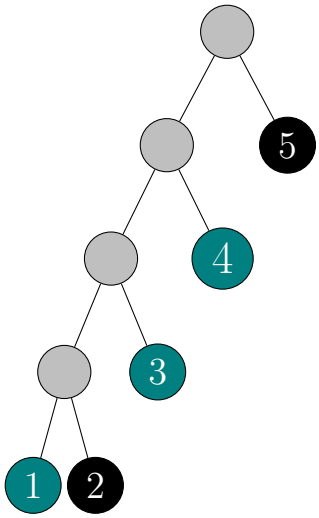

Fig. 7. Binary left caterpillar with five leaves. The right child of each internal node is a leaf. Observe that $\mathrm{triplet}(a, b, c) = 1$ if $a < b < c$. For example, $\mathrm{triplet}(1, 3, 4) = 1$.

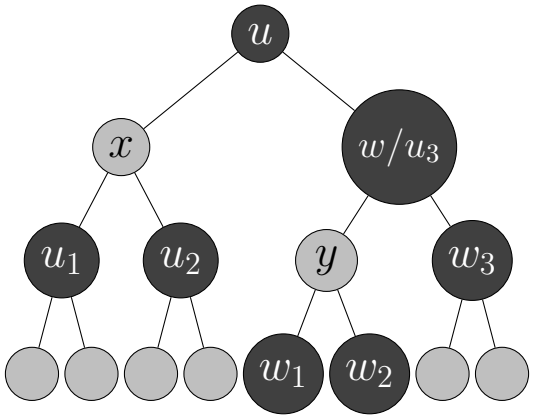

Fig. 8. Binary tree T' constructed based on ternary tree T. Nodes u_1, u_2, u_3 are children of u in ternary tree T. They are leaves in the pattern tree that consists of vertices u, x, u_1, u_2, and u_3. Similarly, vertices w_1, w_2, w_3 are children of w in T. They are leaves in the pattern tree that consists of vertices $w = u_3$, y, w_1, w_2, and w_3.

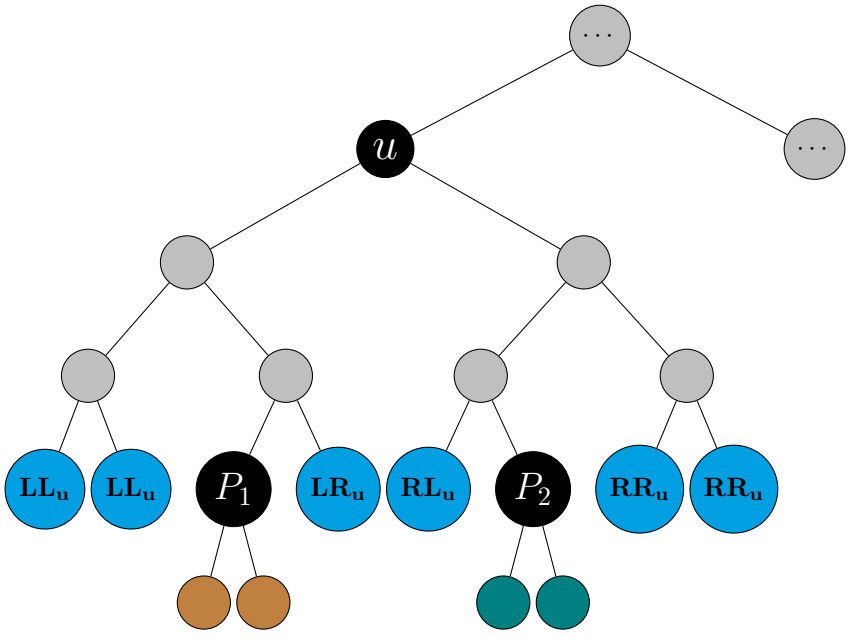

Fig. 9. Algorithm for constructing a coarse solution. Vertices P_1 and P_2 are already processed by the algorithm. The algorithm is currently processing vertex u. It assigns four labels LL_u, LR_u, RL_u, RR_u to yet unlabeled leaves in subtree rooted at u.

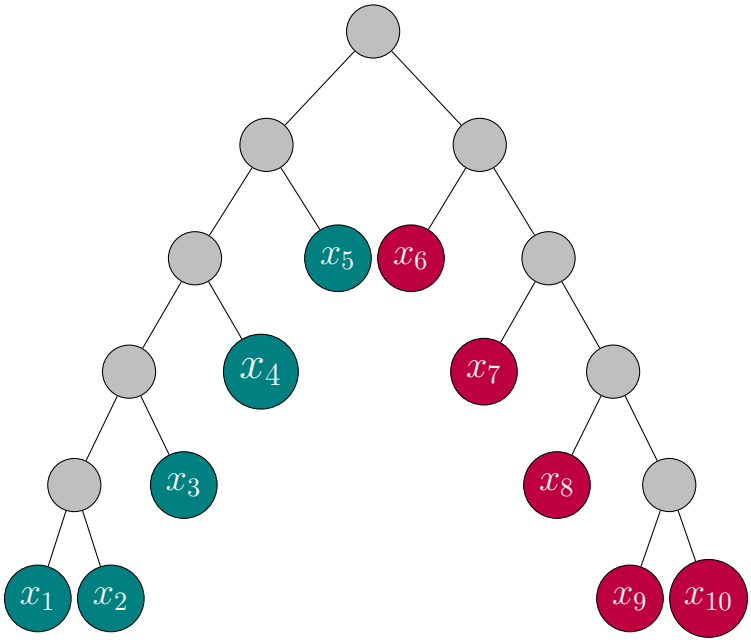

Fig. 10. This phylogenetic predicate consists of patterns obtained from the pattern above by permuting variables $x_1, \ldots, x_{10}$. The predicate requires that at some node u variables $x_1, \ldots, x_{10}$ are split into two equal groups. The first group is assigned to the left subtree; the second group is assigned to the right subtree. Then, the variables in the first group should satisfy the *split-one-to-the-right* constraint, and variables in the second group should satisfy the *split-one-to-the-left* constraint (see Figure 6).

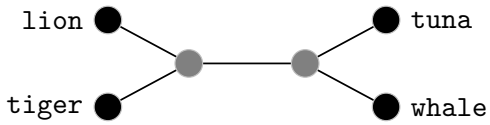

Fig. 11. A quartet tree is the smallest informative unrooted tree used in phylogenetic reconstruction ([34], [72]). Here the quartet {{lion, tiger}, {tuna, whale}} is shown.

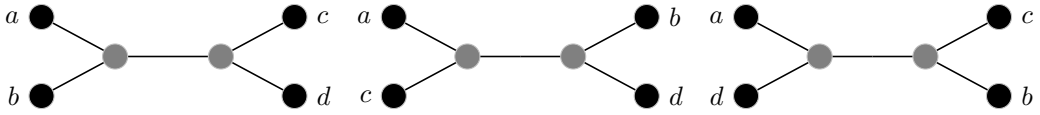

Fig. 12. There are only 3 different (unrooted) quartet trees for items a, b, c, d. The performance of a random assignment achieves a $\frac{1}{3}$-approximation, in expectation.