Reconstructing Phylogenetic Tree From Multipartite Quartet System

Hiroshi Hirai[1]
Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Japan
hirai@mist.i.u-tokyo.ac.jp

Yuni Iwamasa[2]
Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Japan
yuni_iwamasa@mist.i.u-tokyo.ac.jp

Abstract

A phylogenetic tree is a graphical representation of an evolutionary history in a set of taxa in which the leaves correspond to taxa and the non-leaves correspond to speciations. One of important problems in phylogenetic analysis is to assemble a global phylogenetic tree from smaller pieces of phylogenetic trees, particularly, quartet trees. Quartet Compatibility is to decide whether there is a phylogenetic tree inducing a given collection of quartet trees, and to construct such a phylogenetic tree if it exists. It is known that Quartet Compatibility is NP-hard but there are only a few results known for polynomial-time solvable subclasses.

In this paper, we introduce two novel classes of quartet systems, called complete multipartite quartet system and full multipartite quartet system, and present polynomial time algorithms for Quartet Compatibility for these systems. We also see that complete/full multipartite quartet systems naturally arise from a limited situation of block-restricted measurement.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms

Keywords and phrases phylogenetic tree, quartet system, reconstruction

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.57

Acknowledgements We thank Kunihiko Sadakane for bibliographical information.

1 Introduction

A *phylogenetic tree* for finite set $[n] := \{1, 2, \ldots, n\}$ is a tree $T = (V, E)$ such that the set of leaves of T coincides with $[n]$ and each internal node $V \setminus [n]$ has at least three neighbors. A phylogenetic tree represents an evolutionary history in a set of taxa in which the leaves correspond to taxa and the non-leaves correspond to speciations. One of important problems in phylogenetic analysis is to assemble a global phylogenetic tree on $[n]$ (called a *supertree*) from smaller pieces of phylogenetic trees on possibly overlapping subsets of $[n]$; see [17, Section 6].

A *quartet tree* (or *quartet*) is a smallest nontrivial phylogenetic tree, that is, it has four leaves (as taxa) and it is not a star. There are three quartet trees in set $\{a, b, c, d\}$, which are denoted by $ab||cd$, $ac||bd$, and $ad||bc$. Here $ab||cd$ represents the phylogenetic tree such that a and b (c and d) are adjacent to a common node; see Figure 1. Quartet trees are

[1] Supported by KAKENHI Grant Numbers JP26280004, JP17K00029.
[2] Supported by JSPS Research Fellowship for Young Scientists.

29th International Symposium on Algorithms and Computation (ISAAC 2018).
Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 57; pp. 57:1–57:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

used for representing substructures of a (possibly large) phylogenetic tree. A fundamental problem in phylogenetic analysis is to construct a phylogenetic tree having given quartets as substructures. To introduce this problem formally, we need some notations and terminologies. We say that a phylogenetic tree T *displays* a quartet $ab||cd$ if the simple paths connecting a, b and c, d in T, respectively, do not meet, i.e., $ab||cd$ is the "restriction" of T to leaves a, b, c, d; see Figure 2. By a *quartet system* on $[n]$ we mean a collection of quartet trees whose leaves are subsets of $[n]$. We say that T *displays* a quartet system $\mathcal{Q}$ if T displays all quartet trees in $\mathcal{Q}$. A quartet system $\mathcal{Q}$ is said to be *compatible* if there exists a phylogenetic tree displaying $\mathcal{Q}$. Now the problem is formulated as:

Quartet Compatibility

Given: A quartet system $\mathcal{Q}$.

Problem: Determine whether $\mathcal{Q}$ is compatible or not. If it is compatible, obtain a phylogenetic tree T displaying $\mathcal{Q}$.

Quartet Compatibility has been intensively studied in computational biology as well as theoretical computer science, particularly, algorithm design and computational complexity. After a fundamental result by Steel [18] on the NP-hardness of Quartet Compatibility, there have been a large amount of algorithmic results, e.g., efficient heuristics [13, 19], approximation algorithms [3, 4, 12], and parametrized algorithms [7, 10].

In contrast, there are only a few results known for polynomial-time solvable special subclasses:

- Colonius–Schulze [8] established a complete characterization to the abstract quaternary relation N (*neighbors relation*) obtained from a phylogenetic tree T by: $N(a, b, c, d)$ holds if and only if T displays quartet tree $ab||cd$. By using this result, Bandelt–Dress [2] showed that if, for every 4-element set $\{a, b, c, d\}$ of $[n]$, exactly one of $ab||cd$, $ac||bd$, and $ad||cd$ belongs to $\mathcal{Q}$, then Quartet Compatibility for $\mathcal{Q}$ can be solved in polynomial time.
- Aho–Sagiv–Szymanski–Ullman [1] devised a polynomial time algorithm to find a *rooted* phylogenetic tree displaying the input *triple* system. By using this result, Bryant–Steel [5] showed that, if all quartets in $\mathcal{Q}$ have a common label, then Quartet Compatibility for $\mathcal{Q}$ can be solved in polynomial time.

Such results are useful for designing experiments to obtain quartet information from taxa, and also play key roles in developing supertree methods for (incompatible) phylogenetic trees (e.g., [16]).

In this paper, we present two novel tractable classes of quartet systems. To describe our result, we extend the notions of quartets and quartet systems. In addition to $ab||cd$, we consider symbol $ab|cd$ as a quartet, which represents the quartet tree $ab||cd$ or the star with leaves a, b, c, d; see Figure 1. This corresponds to the *weak neighbors relation* in [2, 8], and enables us to capture a degenerate phylogenetic tree in which internal nodes may have degree greater than 3. In a sense, $ab|cd$ means a "possibly degenerate" quartet tree such that the center edge can have zero length. We define that a phylogenetic tree T *displays* $ab|cd$ if the simple paths connecting a, b and c, d in T, respectively, meet at most one node, i.e., the restriction of T to a, b, c, d is $ab||cd$ or star; see Figure 2. Then the concepts of quartet systems, displaying, compatibility, and Quartet Compatibility are naturally extended. A quartet system $\mathcal{Q}$ is said to be *full* on $[n]$ if, for each distinct $a, b, c, d \in [n]$, either one of $ab||cd, ac||bd, ad||bc$ belongs to $\mathcal{Q}$ or all $ab|cd, ac|bd, ad|bc$ belong to $\mathcal{Q}$. The latter situation says that any phylogenetic tree displaying $\mathcal{Q}$ should induce a star on a, b, c, d. Actually the above polynomial-time algorithm by Bandelt–Dress [2] works for full quartet systems.

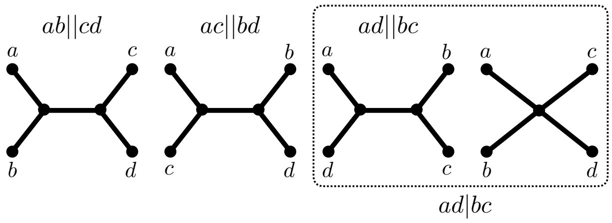

Figure 1 The quartets $ab||cd$, $ac||bd$, and $ad||bc$ represent the first, second, and third phylogenetic trees for a, b, c, d from the left, respectively. $ad|bc$, for example, represents one of the two phylogenetic trees in the dotted curve, that is, $ad||bc$ or the star graph with leaves a, b, c, d.

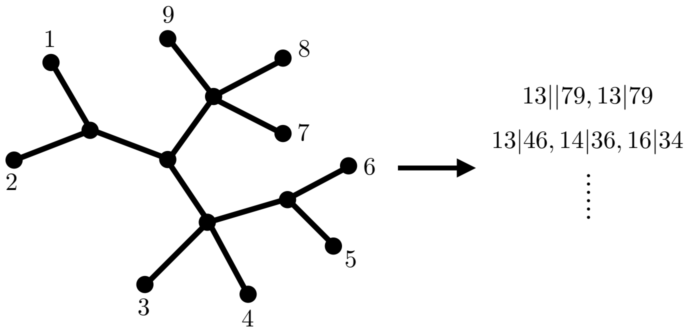

Figure 2 An example of phylogenetic tree T for $\{1, 2, \ldots, 9\}$. T displays, for example, $13||79, 13|79$, and $13|46, 14|36, 16|34$.

Full quartet systems may be viewed as a counter part of complete graphs. We introduce multipartite counterparts for quartet systems. A quartet system $\mathcal{Q}$ is said to be *complete bipartite* relative to bipartition $\{A, B\}$ of $[n]$ with $\min\{|A|, |B|\} \geq 2$ if, for all distinct $a, a' \in A$ and $b, b' \in B$, $\mathcal{Q}$ has exactly one of

$$ab||a'b', \quad ab'||a'b, \quad aa'|bb', \tag{1}$$

and every quartet in $\mathcal{Q}$ is of the above form (1). Note that every phylogenetic tree displays exactly one of three quartets in (1). We next introduce a complete multipartite system. Let $\mathcal{A} := \{A_1, A_2, \ldots, A_r\}$ be a partition of $[n]$ with $|A_i| \geq 2$ for all $i \in [r]$. A quartet system $\mathcal{Q}$ is said to be *complete multipartite* relative to $\mathcal{A}$ or *complete $\mathcal{A}$-partite* if $\mathcal{Q}$ is represented as $\bigcup_{1 \leq i < j \leq r} \mathcal{Q}_{ij}$ for complete bipartite quartet systems $\mathcal{Q}_{ij}$ on $A_i \cup A_j$ with bipartition $\{A_i, A_j\}$. A quartet system $\mathcal{Q}$ is said to be *full multipartite* relative to $\mathcal{A}$ or *full $\mathcal{A}$-partite* if $\mathcal{Q}$ is represented as $\mathcal{Q}_0 \cup \mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_r$, where $\mathcal{Q}_0$ is a complete $\mathcal{A}$-partite quartet system and $\mathcal{Q}_i$ is a full quartet system on A_i for each $i \in [r]$. Our main result is:

▶ **Theorem 1.1.** *If the input quartet system $\mathcal{Q}$ is complete $\mathcal{A}$-partite or full $\mathcal{A}$-partite, then* Quartet Compatibility *can be solved in $O(|\mathcal{A}|n^4)$ time.*

The result for full $\mathcal{A}$-partite quartet systems extends the above polynomial time solvability for full quartet systems by [2]. Also this result has some insights on supertree construction from phylogenetic trees on disjoint groups of taxa. In such a case, we have a full system on each group. Another possible application is given as follows.

Application: Inferring a phylogenetic tree from block-restricted measurements. Quartet-based phylogenetic tree reconstruction methods may be viewed as qualitative approximations of distance methods that construct a phylogenetic tree from (evolutionary) distance $\delta : [n] \times [n] \to \mathbf{R}_+$ among a set $[n]$ of taxa. Here $\mathbf{R}_+$ denotes the set of nonnegative real values. The distance δ naturally gives rise to a full quartet system $\mathcal{Q}$ as follows. Let $\mathcal{Q} := \emptyset$ at first. For all distinct $a, b, c, d \in [n]$, add $ab||cd$ to $\mathcal{Q}$ if $\delta(a,b)+\delta(c,d) < \min\{\delta(a,c)+\delta(b,d), \delta(a,d)+\delta(b,d)\}$. See [9, 14]. Then $\mathcal{Q}$ becomes a full quartet system, after adding $ab|cd$, $ac|bd$, $ac|bd$ if none of $ab||cd$, $ac||bd$, $ac||bd$ belong to $\mathcal{Q}$. If δ coincides with the path-metric of an actual phylogenetic tree T (with nonnegative edge-length), then δ obeys the famous four-point condition on all four elements a, b, c, d [6]:

(4pt) the larger two of $\delta(a,b) + \delta(c,d)$, $\delta(a,c) + \delta(b,d)$, and $\delta(a,d) + \delta(b,c)$ are equal.

In this case, the above definition of quartets matches the neighbors relation of T. Thus, from the full quartet system $\mathcal{Q}$, via the algorithm of [2], we can recover the original phylogenetic tree T (without edge-length).

Next we consider the following limited situation in which complete/full $\mathcal{A}$-partite quartet systems naturally arise. The set $[n]$ of taxa is divided into r groups $A_1, A_2, \ldots, A_r$ (with $|A_i| \geq 2$). By reasons of the cost and/or the difficulty of experiments, we are limited to measure the distance between $a \in A_i$ and $b \in A_j$ via different methods/equipments depending on i, j. Namely we have $\binom{r}{2}$ distance functions $\delta_{ij} : A_i \times A_j \to \mathbf{R}_+$ for $1 \leq i < j \leq r$ but it is meaningless to compare numerical values of δ_{ij} and $\delta_{i'j'}$ for $\{i,j\} \neq \{i',j'\}$. A complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$ is obtained as follows. For distinct i, j, define complete bipartite quartet system $\mathcal{Q}_{ij}$ by: for all distinct $a, a' \in A_i$ and $b, b' \in A_j$ it holds

$$\begin{aligned} ab||a'b' \in \mathcal{Q}_{ij} \quad & \text{if } \delta_{ij}(a,b) + \delta_{ij}(a',b') < \delta_{ij}(a,b') + \delta_{ij}(a',b), \\ ab'||a'b \in \mathcal{Q}_{ij} \quad & \text{if } \delta_{ij}(a,b) + \delta_{ij}(a',b') > \delta_{ij}(a,b') + \delta_{ij}(a',b), \\ aa'|bb' \in \mathcal{Q}_{ij} \quad & \text{if } \delta_{ij}(a,b) + \delta_{ij}(a',b') = \delta_{ij}(a,b') + \delta_{ij}(a',b). \end{aligned}$$

Then $\mathcal{Q} := \bigcup_{1 \leq i < j \leq r} \mathcal{Q}_{ij}$ is a complete $\mathcal{A}$-partite quartet system.

This construction of complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$ is justified as follows. Assume a phylogenetic tree T on $[n]$ with path-metric δ. Assume further that each δ_{ij} is linear on δ, i.e., δ_{ij} is equal to $\alpha_{ij}\delta$ for some unknown constant $\alpha_{ij} > 0$. By (4pt), the situation $\delta_{ij}(a,b) + \delta_{ij}(a',b') < \delta_{ij}(a,b') + \delta_{ij}(a',b)$ implies $\delta(a,b)+\delta(a',b') < \delta(a,b')+\delta(a',b) = \delta(a,a')+\delta(a,b')$, and implies that T displays $ab||a'b'$. The situation $\delta_{ij}(a,b) + \delta_{ij}(a',b') = \delta_{ij}(a,b') + \delta_{ij}(a',b)$ implies $\delta(a,b) + \delta(a',b') = \delta(a,b') + \delta(a',b) \geq \delta(a,a') + \delta(a,b')$, and implies that T displays $aa'|bb'$. Thus, by our algorithm, we can construct a phylogenetic tree T' "similar" to T in the sense that T' and T produce the same result under our limited measurement.

Suppose now that we have additional r distance functions $\delta_i : A_i \times A_i \to \mathbf{R}_+$ for $i \in [r]$. In this case, we naturally obtain a full $\mathcal{A}$-partite quartet system. Indeed, define full quartet system $\mathcal{Q}_i$ on A_i according to δ_i as in the first paragraph. Then $\mathcal{Q} := \bigcup_{1 \leq i < j \leq r} \mathcal{Q}_{ij} \cup \bigcup_{1 \leq i \leq r} \mathcal{Q}_i$ is a full $\mathcal{A}$-partite quartet system to which our algorithm is applicable.

Organization. QUARTET COMPATIBILITY can be viewed as a problem of finding an appropriate laminar family. We first introduce a displaying concept for an arbitrary family of subsets, and then divide QUARTET COMPATIBILITY into two subproblems: The first is to find a family displaying the input quartet system, and the second is to transform the family into a desired laminar family. For the second, we utilize the *laminarization algorithm* developed by Hirai–Iwamasa–Murota–Živný [11] for a completely irrelevant problem in discrete optimization. In Sections 2 and 3, we show the result for complete and full multipartite quartet systems, respectively. The omitted proofs will be given in the full version of this paper.

Preliminaries. A family $\mathcal{L} \subseteq 2^{[n]}$ is said to be *laminar* if $X \subseteq Y$, $X \supseteq Y$, or $X \cap Y = \emptyset$ holds for all $X, Y \in \mathcal{L}$. A phylogenetic tree can be encoded into a laminar family as follows. Let $T = (V, E)$ be a phylogenetic tree for $[n]$. By deleting internal edge $e \in E$, the tree T is separated into two connected components, and so is $[n]$. We denote by $\{X_e, Y_e\}$ the bipartition induced by e. By choosing either X_e or Y_e appropriately for each internal edge $e \in E$, we can construct a laminar family $\mathcal{L}$ on $[n]$ with $\min\{|X|, |[n] \setminus X|\} \geq 2$ for all $X \in \mathcal{L}$. Conversely, let $\mathcal{L}$ on $[n]$ be a laminar family with $\min\{|X|, |[n] \setminus X|\} \geq 2$ for all $X \in \mathcal{L}$. Then we construct the set $\hat{\mathcal{L}} := \{\{X, [n] \setminus X\} \mid X \in \mathcal{L}\}$ of bipartitions from $\mathcal{L}$. It is known [6] that, for such $\hat{\mathcal{L}}$, there uniquely exists a phylogenetic tree that induces $\hat{\mathcal{L}}$.

2 Complete multipartite quartet system

2.1 Displaying and Laminarization

In this subsection, we explain that Quartet Compatibility for complete multipartite quartet systems can be divided into two subproblems named as Displaying and Laminarization. Let $\mathcal{A} := \{A_1, A_2, \ldots, A_r\}$ be a partition of $[n]$ with $|A_i| \geq 2$ for all $i \in [r]$, and $\mathcal{Q}$ be a complete $\mathcal{A}$-partite quartet system. We say that a family $\mathcal{F} \subseteq 2^{[n]}$ *displays* $\mathcal{Q}$ if, for all distinct $i, j \in [r]$, $a, a' \in A_i$, and $b, b' \in A_j$,

$$ab||a'b' \in \mathcal{Q} \iff \text{there is } X \in \mathcal{F} \text{ satisfying } a, b \in X \not\ni a', b' \text{ or } a, b \notin X \ni a', b'.$$

We can easily see that, if $\mathcal{L}$ is laminar, then $\mathcal{L}$ displays exactly one complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$. Furthermore, such $\mathcal{Q}$ is the same as the one displayed by the phylogenetic tree corresponding to $\mathcal{L}$. Thus Quartet Compatibility for a complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$ can be viewed as the problem finding a laminar family $\mathcal{L}$ displaying $\mathcal{Q}$ if it exists.

It can happen that different families may display the same complete $\mathcal{A}$-partite quartet system. To cope with such complications, we define an equivalence relation $\sim$ on sets $X, Y \subseteq [n]$ by: $X \sim Y$ if $\{X\}$ and $\{Y\}$ display the same complete $\mathcal{A}$-partite quartet system. Let $[X] := \{Y \subseteq [n] \mid X \sim Y\}$ for $X \subseteq [n]$. A set $X \subseteq [n]$ is called an $\mathcal{A}$-*cut* if $X \not\sim \emptyset$, i.e., $X \notin [\emptyset]$. For $X \subseteq [n]$, define

$$\langle X \rangle := \bigcup \{A_i \in \mathcal{A} \mid \emptyset \neq X \cap A_i \neq A_i\}. \tag{2}$$

One can see that X is an $\mathcal{A}$-cut if and only if $\emptyset \neq X \cap A_i \neq A_i$ holds for at least two $i \in [r]$, i.e., $\langle X \rangle \supseteq A_i \cup A_j$ for some distinct $i, j \in [r]$. We consider only $\mathcal{A}$-cuts if the input quartet system $\mathcal{Q}$ is complete $\mathcal{A}$-partite. Indeed, let $\mathcal{F}$ be a family and $\mathcal{F}'$ the $\mathcal{A}$-cut family in $\mathcal{F}$. Then both $\mathcal{F}$ and $\mathcal{F}'$ display the same complete $\mathcal{A}$-partite quartet system.

One can see that, for $\mathcal{A}$-cuts X, Y, it holds that $X \sim Y \Leftrightarrow \{\langle X \rangle \cap X, \langle X \rangle \setminus X\} = \{\langle Y \rangle \cap Y, \langle Y \rangle \setminus Y\}$. The equivalence relation is naturally extended to $\mathcal{A}$-cut families $\mathcal{F}, \mathcal{G}$ by: $\mathcal{F} \sim \mathcal{G} \Leftrightarrow \mathcal{F}/\!\sim\, = \mathcal{G}/\!\sim$, where $\mathcal{F}/\!\sim\, := \{[X] \mid X \in \mathcal{F}\}$. It is clear, by the definition of $\sim$, that if $\mathcal{F} \sim \mathcal{G}$ then both $\mathcal{F}$ and $\mathcal{G}$ display the same complete $\mathcal{A}$-partite quartet system. An $\mathcal{A}$-cut family $\mathcal{F}$ is said to be *laminarizable* if there is a laminar family $\mathcal{L}$ with $\mathcal{F} \sim \mathcal{L}$.

By the above argument, Quartet Compatibility for a complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$ can be divided into the following two subproblems: (i) if $\mathcal{Q}$ is compatible, then find a laminarizable family $\mathcal{F}$ displaying $\mathcal{Q}$, and (ii) if $\mathcal{F}$ is laminarizable, then find a laminar family $\mathcal{L}$ with $\mathcal{L} \sim \mathcal{F}$. (i) and (ii) can be formulated as Displaying and Laminarization, respectively.

Displaying
Given: A complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$.
Problem: Either detect the incompatibility of $\mathcal{Q}$, or obtain some $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$. In addition, if $\mathcal{Q}$ is compatible, then $\mathcal{F}$ should be laminarizable.

Laminarization
Given: An $\mathcal{A}$-cut family $\mathcal{F}$.
Problem: Determine whether $\mathcal{F}$ is laminarizable or not. If $\mathcal{F}$ is laminarizable, obtain a laminar $\mathcal{A}$-cut family $\mathcal{L}$ with $\mathcal{L} \sim \mathcal{F}$.

Here, in LAMINARIZATION, we assume that no distinct X, Y with $X \sim Y$ are contained in $\mathcal{F}$, i.e., $|\mathcal{F}| = |\mathcal{F}/\sim|$.

QUARTET COMPATIBILITY for complete multipartite quartet systems can be solved as follows.

- Suppose that $\mathcal{Q}$ is compatible. First, by solving DISPLAYING, we obtain a laminarizable $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$. Then, by solving LAMINARIZATION for $\mathcal{F}$, we obtain a laminar $\mathcal{A}$-cut family $\mathcal{L}$ with $\mathcal{L} \sim \mathcal{F}$. Since $\mathcal{L} \sim \mathcal{F}$, $\mathcal{L}$ also displays $\mathcal{Q}$.
- Suppose that $\mathcal{Q}$ is not compatible. By solving DISPLAYING, we can detect the incompatibility of $\mathcal{Q}$ or we obtain some $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$. In the former case, we are done. In the latter case, by solving LAMINARIZATION for $\mathcal{F}$, we can detect the non-laminarizability of $\mathcal{F}$, which implies the incompatibility of $\mathcal{Q}$.

In [11], the authors presented an $O(n^4)$-time algorithm for LAMINARIZATION.

▶ **Theorem 2.1** ([11]). LAMINARIZATION *can be solved in* $O(n^4)$ *time.*

In Section 2.3, we give an $O(rn^4)$-time algorithm for DISPLAYING (Theorem 2.8). Thus, by Theorems 2.1 and 2.8, we obtain Theorem 1.1 for complete $\mathcal{A}$-partite quartet systems.

2.2 Algorithm for complete bipartite quartet system

We first construct a polynomial time algorithm for QUARTET COMPATIBILITY for complete bipartite quartet systems. In the following, $\mathcal{A}$ is a bipartition of $[n]$ represented as $\{A, B\}$ with $\min\{|A|, |B|\} \geq 2$. Note that X is an $\mathcal{A}$-cut if and only if $\emptyset \neq X \cap A \neq A$ and $\emptyset \neq X \cap B \neq B$, and that $X \sim Y$ if and only if $X = Y$ or $X = [n] \setminus Y$.

Choose an arbitrary $a \in [n]$. For a compatible bipartite quartet system $\mathcal{Q}$, there is a laminar $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ such that there is no $X \in \mathcal{F}$ with $a \in X$. The following proposition implies that such $\mathcal{F}$ is unique.

▶ **Proposition 2.2.** *Suppose that a bipartite quartet system* $\mathcal{Q}$ *is compatible. Then a laminarizable* $\mathcal{A}$*-cut family* $\mathcal{F}$ *displaying* $\mathcal{Q}$ *is uniquely determined up to* $\sim$.

We introduce two notations used in Sections 2.2.1 and 2.2.2. For $\mathcal{F} \subseteq 2^{[n]}$ and $X \subseteq [n]$, we denote $\{F \cup X \mid F \in \mathcal{F}\}$ by $\mathcal{F} \sqcup X$. For $C \subseteq A$ and $D \subseteq B$, we denote by $\mathcal{Q}|_{C,D}$ the set of quartet trees for c, c', d, d' ($c, c' \in C$ and $d, d' \in D$) in $\mathcal{Q}$.

2.2.1 Case of $|A| = 2$ or $|B| = 2$

We consider the case of $|A| = 2$ or $|B| = 2$. Without loss of generality, we assume $A = \{a_0, a\}$ with $a_0 \neq a$.

We first explain the idea behind our algorithm (Algorithm 1). Assume that a complete $\{\{a_0, a\}, B\}$-partite quartet system $\mathcal{Q}$ is compatible. By Proposition 2.2, there uniquely exists a laminar $\{\{a_0, a\}, B\}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ such that no $X \in \mathcal{F}$ contains a_0.

This implies that all $X \in \mathcal{F}$ contains a since $\mathcal{F}$ is an $\{\{a_0, a\}, B\}$-cut family. Hence, by the laminarity, $\mathcal{F}$ is a chain $\{B_1, B_2, \ldots, B_m\} \sqcup \{a\}$ with $\emptyset =: B_0 \subsetneq B_1 \subsetneq B_2 \subsetneq \cdots \subsetneq B_m \subsetneq B_{m+1} := B$.

Choose an arbitrary $b \in B$. Consider the index $k \in [m+1]$ such that $b \in B_k$ and $b \notin B_{k-1}$. Partition B into three sets $B^- := B_{k-1}$, $B^= := B_k \setminus B_{k-1}$, and $B^+ := B \setminus B_k$. The tripartition $\{B^-, B^=, B^+\}$ can be determined by checking quartets in $\mathcal{Q}$ having leaves a_0, a, b:

$$b' \in B^- \iff a_0 b || a b' \in \mathcal{Q}, \tag{3}$$
$$b' \in B^= \iff b' = b \text{ or } a_0 a | b b' \in \mathcal{Q}, \tag{4}$$
$$b' \in B^+ \iff a_0 b' || a b \in \mathcal{Q}. \tag{5}$$

Observe that $\mathcal{Q}|_{\{a_0,a\},B^-}$ is displayed by $\mathcal{F}^- := \{B_1, \ldots, B_{k-2}\} \sqcup \{a\}$ and that $\mathcal{Q}|_{\{a_0,a\},B^+}$ is displayed by $\mathcal{F}^+ := \{B_{k+1} \setminus B_k, \ldots, B_m \setminus B_k\} \sqcup \{a\}$. After determining $B^-, B^=, B^+$, we can apply recursively the same procedure to $\mathcal{Q}|_{\{a_0,a\},B^-}$ and $\mathcal{Q}|_{\{a_0,a\},B^+}$, and obtain $\mathcal{F}^-$ and $\mathcal{F}^+$. Combining them with $B_k = B^= \cup B^-$ and $B_{k-1} = B^-$, we obtain $\mathcal{F} = \{B_1, B_2, \ldots, B_m\} \sqcup \{a\}$ as required.

The formal description of Algorithm 1 is the following:

Algorithm 1 (for complete $\{\{a_0, a\}, B\}$-partite quartet system with pivot a).

Input: A complete $\{\{a_0, a\}, B\}$-partite quartet system $\mathcal{Q}$.

Output: Either detect the incompatibility of $\mathcal{Q}$, or obtain the (unique) laminar $\{\{a_0, a\}, B\}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ such that no $X \in \mathcal{F}$ contains a_0.

Step 1: If $\mathcal{Q} = \emptyset$, that is, $|B|$ is at most one, then output the emptyset and stop.

Step 2: Choose an arbitrary $b \in B$. Define B^-, $B^=$, and B^+ as (3), (4), and (5), respectively.

Step 3: If Algorithm 1 for $\mathcal{Q}|_{\{a_0,a\},B^+}$ with pivot a detects the incompatibility of $\mathcal{Q}|_{\{a_0,a\},B^+}$ or Algorithm 1 for $\mathcal{Q}|_{\{a_0,a\},B^-}$ with pivot a detects the incompatibility of $\mathcal{Q}|_{\{a_0,a\},B^-}$, then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, let $\mathcal{F}^+$ and $\mathcal{F}^-$ be the output families of Algorithm 1 for $\mathcal{Q}|_{\{a_0,a\},B^+}$ and for $\mathcal{Q}|_{\{a_0,a\},B^-}$, respectively. Define

$$\mathcal{F} := \mathcal{F}^- \cup \left(\mathcal{F}^+ \sqcup (B^- \cup B^=)\right) \cup \left(\left(\{B^-, B^- \cup B^=\} \setminus \{\emptyset, B\}\right) \sqcup \{a\}\right).$$

Step 4: If $\mathcal{F}$ displays $\mathcal{Q}$, then output $\mathcal{F}$. Otherwise, output "$\mathcal{Q}$ is not compatible."

▶ **Proposition 2.3.** *Algorithm 1 solves* Quartet Compatibility *for a complete* $\{\{a_0, a\}, B\}$*-partite quartet system* $\mathcal{Q}$ *in* $O(|\mathcal{Q}|)$ *time.*

2.2.2 General case

We consider general complete bipartite quartet systems; $\mathcal{A}$ is a bipartition $\{A, B\}$ of $[n]$. As in Section 2.2.1, we first explain the idea behind our algorithm (Algorithm 2). Assume that a complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$ is compatible. By Proposition 2.2, there uniquely exists a laminar $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ such that no $X \in \mathcal{F}$ contains a_0.

Define $\mathcal{F}^a$ as the output of Algorithm 1 for $\mathcal{Q}|_{\{a_0,a\},B}$ with pivot a. Since $\mathcal{Q}|_{\{a_0,a\},B}$ is displayed by $\{X \cap B \mid a \in X \in \mathcal{F}\} \sqcup \{a\}$, it holds that $\mathcal{F}^a = \{X \cap B \mid a \in X \in \mathcal{F}\} \sqcup \{a\}$ by Propositions 2.2 and 2.3. Define $\mathcal{F} \cap B := \{X \cap B \mid X \in \mathcal{F}\}$. It can be easily seen that $\mathcal{F} \cap B = \bigcup_{a \in A \setminus \{a_0\}} \{X \cap B \mid X \in \mathcal{F}^a\}$. In the following, we consider to combine $\mathcal{F}^a$s appropriately.

Take any $D \in \mathcal{F} \cap B$, and define $A_D := \{a \in A \setminus \{a_0\} \mid \{a\} \cup D \in \mathcal{F}^a\}$. By the laminarity of $\mathcal{F}$, $A_D \cup D$ is the unique maximal set X in $\mathcal{F}$ such that $X \cap B = D$. Hence we can construct the set $\mathcal{G} := \{A_D \cup D \mid D \in \mathcal{F} \cap B\} \subseteq \mathcal{F}$ from $\mathcal{F}^a$ $(a \in A \setminus \{a_0\})$. Note that $\mathcal{G}$ is laminar.

All the left is to determine all nonmaximal sets $X \in \mathcal{F}$ with $X \cap B = D$ for each $D \in \mathcal{F} \cap B$. Fix an arbitrary $D \in \mathcal{F} \cap B$. Observe that, by the laminarity of $\mathcal{F}$, the set $\{X \in \mathcal{F} \mid X \cap B = D\}$ is a chain $\{X_1, X_2, \dots, X_m\}$ with $X_1 \subsetneq X_2 \subsetneq \cdots \subsetneq X_m = A_D \cup D$. We are going to identify this chain with the help of Algorithm 1. Let $X^- := \bigcup\{X' \in \mathcal{G} \mid X' \subsetneq X_m\}$, and choose an arbitrary $b_0 \in B \setminus D$ and $b \in D$. Note that $X_1 \supseteq X^-$ by the laminarity of $\mathcal{F}$. We first consider the easier case $X_1 \cap A \supsetneq X^- \cap A$. Then apply Algorithm 1 to $\mathcal{Q}|_{A_D \setminus X^-, \{b_0, b\}}$ and obtain $\{(X_1 \setminus X^-) \cap A, (X_2 \setminus X^-) \cap A, \dots, (X_m \setminus X^-) \cap A\} \sqcup \{b_0, b\}$ (that displays $\mathcal{Q}|_{A_D \setminus X^-, \{b_0, b\}}$). From this we obtain $\{X_1, X_2, \dots, X_m\}$, as required.

Next consider the case $X_1 \cap A = X^- \cap A$. In this case, by applying Algorithm 1 to $\mathcal{Q}|_{A_D \setminus X^-, \{b_0, b\}}$, we only obtain $\{(X_2 \setminus X^-) \cap A, \dots, (X_m \setminus X^-) \cap A\} \sqcup \{b_0, b\}$, and hence $\{X_2, \dots, X_m\}$. Therefore we need to construct X_1 individually as follows. Pick any $a \in X^- \cap A$ and retake b from $D \setminus X'$ for maximal $X' \in \mathcal{G}$ with $a \in X' \subseteq X^-$. For $a' \in (X_m \setminus X^-) \cap A$, it cannot happen that $ab_0||a'b \in \mathcal{Q}$ since all $X \in \mathcal{F}$ containing a', b also include a. Furthermore we can say that $ab||a'b_0 \in \mathcal{Q}$ if and only if $a' \notin X_1 (\ni a, b)$. This implies that $aa'|bb_0 \in \mathcal{Q}$ if and only if a' belongs to X_1. Hence it holds that X_1 is the union of $X^- \cup D$ and all elements $a' \in A_D \setminus X^-$ with $aa'|b_0 b \in \mathcal{Q}$.

The formal description of Algorithm 2 is the following; note that, if $\mathcal{F}$ is laminar, then $|\mathcal{F}|$ is at most $2n$ (see e.g., [15, Theorem 3.5]).

▶ **Proposition 2.4.** *Algorithm 2 solves* Quartet Compatibility *for a complete bipartite quartet system $\mathcal{Q}$ in $O(|\mathcal{Q}|)$ time.*

2.3 Algorithm for complete multipartite quartet system

In this subsection, we present a polynomial time algorithm for complete multipartite quartet systems. First we introduce some notations before giving the outline of our proposed algorithm (Algorithm 4). Let $\mathcal{A} := \{A_1, A_2, \dots, A_r\}$ be a partition of $[n]$ with $|A_i| \geq 2$ for all $i \in [r]$. For the analysis of the running-time of Algorithm 4, we assume $|A_1| \geq |A_2| \geq \cdots \geq |A_r|$. For $R \subseteq [r]$ with $|R| \geq 2$, let $\mathcal{A}_R := \{A_i\}_{i \in R}$ and $A_R := \bigcup_{i \in R} A_i$. For complete $\mathcal{A}$-partite quartet system $\mathcal{Q} = \bigcup_{1 \leq i < j \leq r} \mathcal{Q}_{ij}$, define $\mathcal{Q}_R := \bigcup_{i,j \in R, i<j} \mathcal{Q}_{ij}$. That is, $\mathcal{Q}_R$ is the complete $\mathcal{A}_R$-partite quartet system included in $\mathcal{Q}$. For $\mathcal{A}$-cut family $\mathcal{F}$, define $\mathcal{F}_R := \{X \cap A_R \mid X \in \mathcal{F}$ such that $X \cap A_R$ is an $\mathcal{A}_R$-cut$\}$. Note that $\mathcal{F}_R$ is an $\mathcal{A}_R$-cut family. Then we can easily see the following lemma, which says that partial information $\mathcal{F}_R$ of $\mathcal{F}$ can be obtained from $\mathcal{Q}_R$.

▶ **Lemma 2.5.** *Suppose $R \subseteq [r]$ with $|R| \geq 2$. If $\mathcal{Q}$ is displayed by $\mathcal{F}$, then $\mathcal{Q}_R$ is displayed by $\mathcal{F}_R$. Furthermore, if $\mathcal{Q}$ is compatible, then so is $\mathcal{Q}_R$.*

Our algorithm for Displaying is to construct an $\mathcal{A}_{[t]}$-cut family $\mathcal{F}_t$ displaying $\mathcal{Q}_{[t]}$ for $t = 2, 3, \dots, r$ in turn as follows.

- First we obtain an $\mathcal{A}_{\{1,2\}}$-cut family $\mathcal{F}_2$ displaying $\mathcal{Q}_{12}$ by Algorithm 2.
- For $t \geq 2$, we can extend an $\mathcal{A}_{[t-1]}$-cut family $\mathcal{F}_{t-1}$ displaying $\mathcal{Q}_{[t-1]}$ to an $\mathcal{A}_{[t]}$-cut family $\mathcal{F}_t$ displaying $\mathcal{Q}_{[t]}$ by Algorithm 3. In order to construct $\mathcal{F}_t$ in Algorithm 3, we use an $\mathcal{A}_{\{i,t\}}$-cut family $\mathcal{G}_i$ displaying $\mathcal{Q}_{it}$ for all $i \in [t-1]$. These $\mathcal{G}_i$ can be obtained by Algorithm 2.
- We perform the above extension step for $t = 3$ to $t = r$, and then obtain a desired $\mathcal{A}$-cut family $\mathcal{F} := \mathcal{F}_r$. This is described in Algorithm 4.

Algorithm 2 (for complete bipartite quartet system).

Input: A complete bipartite quartet system $\mathcal{Q}$.

Output: Either detect the incompatibility of $\mathcal{Q}$, or obtain a laminar $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$.

Step 1: Fix an arbitrary $a_0 \in A$. For each $a \in A \setminus \{a_0\}$, we execute Algorithm 1 for $\mathcal{Q}|_{\{a_0,a\},B}$ with pivot a. If Algorithm 1 outputs "$\mathcal{Q}|_{\{a_0,a\},B}$ is not compatible" for some a, then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, obtain the output $\mathcal{F}^a$ for each a.

Step 2: Let $\mathcal{G} := \emptyset$. For each $a \in A \setminus \{a_0\}$, update $\mathcal{G}$ as

$$\begin{aligned}\mathcal{G} \leftarrow \{X \in \mathcal{F}^a \mid \nexists Y \in \mathcal{G} \text{ such that } X \cap B = Y \cap B\} \\ \cup \{Y \in \mathcal{G} \mid \nexists X \in \mathcal{F}^a \text{ such that } X \cap B = Y \cap B\} \\ \cup (\{Y \in \mathcal{G} \mid \exists X \in \mathcal{F}^a \text{ such that } X \cap B = Y \cap B\} \sqcup \{a\}).\end{aligned}$$

If $|\mathcal{G}| > 2n$ for some a, then output "$\mathcal{Q}$ is not compatible" and stop.

Step 3: If $\mathcal{G}$ is not laminar, then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, define $\mathcal{F} := \mathcal{G}$. For each $X \in \mathcal{G}$, do the following:

3-1: Let $X^- := \bigcup\{X' \in \mathcal{G} \mid X' \subsetneq X\}$, and choose an arbitrary $b_0 \in B \setminus X$ and $b \in X \cap B$.

3-2: Execute Algorithm 1 for $\mathcal{Q}|_{(X\setminus X^-)\cap A,\{b_0,b\}}$ with pivot b. If Algorithm 1 outputs "$\mathcal{Q}|_{(X\setminus X^-)\cap A,\{b_0,b\}}$ is not compatible," then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, define

$$\mathcal{H} := \text{the output family of Algorithm 1} \sqcup (X^- \cup (X \cap B)).$$

If $X^- \neq \emptyset$, then go to Step 3-3. Otherwise, go to Step 3-4

3-3: Choose an arbitrary $a \in X^- \cap A$ and retake b from $(X \setminus X') \cap B$ for maximal $X' \in \mathcal{G}$ with $a \in X' \subseteq X^-$. Define $X_1 := X^- \cup (X \cap B) \cup \{a' \in (X \setminus X^-) \cap A \mid aa'|b_0 b \in \mathcal{Q}\}$. If X_1 is not included in the minimal element in $\mathcal{H}$, then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, update $\mathcal{H} \leftarrow \mathcal{H} \cup \{X_1\}$.

3-4: $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{H}$.

Step 4: If $\mathcal{F}$ displays $\mathcal{Q}$, then output $\mathcal{F}$. Otherwise, output "$\mathcal{Q}$ is not compatible."

As a compatible complete bipartite quartet system (Proposition 2.2), a compatible complete multipartite quartet system $\mathcal{Q}$ induces some kind of uniqueness of a laminarizable family displaying $\mathcal{Q}$, which ensures the validity of our proposed algorithm.

▶ **Proposition 2.6.** *Suppose that a complete $\mathcal{A}$-partite quartet system $\mathcal{Q}$ is compatible. Then a minimal laminarizable $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ is uniquely determined up to $\sim$.*

Algorithm 3 constructs a minimal laminarizable family $\mathcal{F}_t$ displaying $\mathcal{Q}_{[t]}$ from a minimal laminarizable family $\mathcal{F}_{t-1}$ displaying $\mathcal{Q}_{[t-1]}$. We define a partial order relation $\prec$ in $\mathcal{A}$-cuts by: $X \prec Y$ if $\langle X \rangle \subsetneq \langle Y \rangle$ and $\{\langle X \rangle \cap X, \langle X \rangle \setminus X\} = \{\langle X \rangle \cap Y, \langle X \rangle \setminus Y\}$. Define $X \preceq Y$ by $X \prec Y$ or $X = Y$. For nonempty $R \subseteq [r]$, we define $\sim_R$ for $\mathcal{A}$-cuts by:

$$X \sim_R Y \iff \{\langle X \rangle_R \cap X, \langle X \rangle_R \setminus X\} = \{\langle Y \rangle_R \cap Y, \langle Y \rangle_R \setminus Y\},$$

where $\langle X \rangle_R := \langle X \rangle \cap A_R$ and $\langle Y \rangle_R := \langle Y \rangle \cap A_R$; recall (2) for the notation $\langle X \rangle$. We abbreviate $\{i_1, i_2, \ldots, i_k\}$ as $i_1 i_2 \cdots i_k$ for distinct $i_1, i_2, \ldots, i_k$. It is noted that, if $\mathcal{F}$ is laminarizable and $X \not\sim Y$ for all distinct $X, Y \in \mathcal{F}$, then $|\mathcal{F}|$ is at most $2n = 2|A_{[r]}|$.

The following proposition shows that Algorithm 3 actually works.

Algorithm 3 (for extending $\mathcal{F}'$ to $\mathcal{F}$).

Input: An $\mathcal{A}$-cut family $\mathcal{F}'$ with $|\mathcal{F}'| \leq 2|A_{[r-1]}|$ displaying $\mathcal{Q}_{[r-1]}$.

Output: Either detect the incompatibility of $\mathcal{Q}$, or obtain $\mathcal{A}$-cut family $\mathcal{F}$ with $|\mathcal{F}| \leq 2n = 2|A_{[r]}|$ displaying $\mathcal{Q}$.

Step 1: For each $i \in [r-1]$, execute Algorithm 2 for $\mathcal{Q}_{ir}$. If Algorithm 2 returns "$\mathcal{Q}_{ir}$ is not compatible" for some $i \in [r-1]$, then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, obtain $\mathcal{G}_i$ for all $i \in [r-1]$. Let $\mathcal{F} := \emptyset$.

Step 2: If $\mathcal{F}' = \emptyset$, update as $\mathcal{F} \leftarrow \mathcal{F} \cup \bigcup_{i\in[r-1]} \mathcal{G}_i$, and go to Step 3. Otherwise, do the following: Take any $X' \in \mathcal{F}'$. Let $\{i_1, i_2, \ldots, i_k\}$ be the set of indices $i \in [r-1]$ with $\langle X' \rangle = A_{i_1 i_2 \ldots i_k}$. Let $\mathcal{F}^{X'}$ be the set of maximal $\mathcal{A}$-cuts Y with respect to $\prec$ such that
- there is $R \subseteq \{i_1, i_2, \ldots, i_k\}$ with $\langle Y \rangle = A_{R\cup\{r\}}$ and $Y \sim_R X'$, and
- there are $X_i \in \mathcal{G}_i$ with $Y \sim_{ir} X_i$ for all $i \in R$.

Then update as $\mathcal{F} \leftarrow \mathcal{F} \cup \{X'\} \cup \mathcal{F}^{X'}$ and $\mathcal{F}' \leftarrow \mathcal{F}' \setminus \{X'\}$, and go to Step 2.

Step 3: Update as

$$\mathcal{F} \leftarrow \text{the set of maximal elements in } \mathcal{F} \text{ with respect to } \prec.$$

If $|\mathcal{F}| \leq 2n$, then output $\mathcal{F}$. Otherwise, output "$\mathcal{Q}$ is not compatible."

▶ **Proposition 2.7.** *If Algorithm 3 outputs $\mathcal{F}$, then $\mathcal{F}$ displays $\mathcal{Q}$. In addition, if $\mathcal{Q}$ is compatible and $\mathcal{F}'$ is a minimal laminarizable $\mathcal{A}_{[r-1]}$-cut family displaying $\mathcal{Q}_{[r-1]}$, then $\mathcal{F}$ is a minimal laminarizable $\mathcal{A}$-cut family.*

Our proposed algorithm for Displaying is the following.

Algorithm 4 (for Displaying).

Step 1: Execute Algorithm 2 for $\mathcal{Q}_{12}$. If Algorithm 2 returns "$\mathcal{Q}_{12}$ is not compatible," then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, obtain $\mathcal{F}_2$.

Step 2: For $t = 3, \ldots, r$, execute Algorithm 3 for $\mathcal{F}_{t-1}$. If Algorithm 3 returns "$\mathcal{Q}_{[t]}$ is not compatible," then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, obtain $\mathcal{F}_t$.

Step 3: Output $\mathcal{F} := \mathcal{F}_r$.

▶ **Theorem 2.8.** *Algorithm 4 solves Displaying in $O(rn^4)$ time. Furthermore, if the input is compatible, then the output is a minimal laminarizable $\mathcal{A}$-cut family.*

3 Full multipartite quartet system

3.1 Full Displaying and Full Laminarization

As in Section 2.1, we see that Quartet Compatibility for full multipartite quartet systems can be divided into two subproblems named as Full Displaying and Full Laminarization. The outline of the argument is the same as the case of complete multipartite quartet systems in Section 2.1. We say that a family $\mathcal{F} \subseteq 2^{[n]}$ *displays* a full quartet system $\mathcal{Q}$ on finite set $A \subseteq [n]$ if for all distinct $a, b, c, d \in A$,

$$ab||cd \in \mathcal{Q} \iff \text{there is } X \in \mathcal{F} \text{ satisfying } a, b \in X \not\ni c, d \text{ or } a, b \notin X \ni c, d.$$

Let $\mathcal{A} := \{A_1, A_2, \ldots, A_r\}$ be a partition of $[n]$ with $|A_i| \geq 2$ for all $i \in [r]$. We also say that $\mathcal{F}$ *displays* a full $\mathcal{A}$-partite quartet system $\mathcal{Q} = \mathcal{Q}_0 \cup \mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_r$, where $\mathcal{Q}_0$ is complete

$\mathcal{A}$-partite and $\mathcal{Q}_i$ is full on A_i for each $i \in [r]$, if $\mathcal{F}$ displays all $\mathcal{Q}_0, \mathcal{Q}_1, \ldots, \mathcal{Q}_r$. Thus QUARTET COMPATIBILITY for full $\mathcal{A}$-partite quartet system $\mathcal{Q}$ can also be viewed as the problem of finding a laminar family $\mathcal{L}$ displaying $\mathcal{Q}$ if it exists.

We also introduce an equivalent relation $\approx$ on sets $X, Y \subseteq [n]$ by: $X \approx Y$ if the families $\{X\}$ and $\{Y\}$ display the same full $\mathcal{A}$-partite quartet system. A set $X \subseteq [n]$ is called a *weak $\mathcal{A}$-cut* if $X \not\approx \emptyset$. One can see that X is a weak $\mathcal{A}$-cut if and only if X is an $\mathcal{A}$-cut, or $\langle X \rangle = A_i$ for some $i \in [r]$ and $\min\{|X|, |A_i \setminus X|\} \geq 2$. One can see that, for weak $\mathcal{A}$-cuts X, Y, it holds that $X \approx Y \Leftrightarrow \{\langle X \rangle \cap X, \langle X \rangle \setminus X\} = \{\langle Y \rangle \cap Y, \langle Y \rangle \setminus Y\}$. The equivalence relation is extended to weak $\mathcal{A}$-cut families $\mathcal{F}, \mathcal{G}$ by: $\mathcal{F} \approx \mathcal{G} \Leftrightarrow \mathcal{F}/{\approx} = \mathcal{G}/{\approx}$, where $\mathcal{F}/{\approx}$ is defined as in Section 2.1. A weak $\mathcal{A}$-cut family $\mathcal{F}$ is said to be *laminarizable* if there is a laminar family $\mathcal{L}$ with $\mathcal{F} \approx \mathcal{L}$. Note that an $\mathcal{A}$-cut is a weak $\mathcal{A}$-cut, and for $\mathcal{A}$-cuts or $\mathcal{A}$-cut families, the equivalence relations $\sim$ and $\approx$ are the same.

By the same argument as in Section 2.1, QUARTET COMPATIBILITY for a full $\mathcal{A}$-partite quartet system $\mathcal{Q}$ can be divided into the following two subproblems.

Full Displaying
Given: A full $\mathcal{A}$-partite quartet system $\mathcal{Q}$.
Problem: Either detect the incompatibility of $\mathcal{Q}$, or obtain some weak $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$. In addition, if $\mathcal{Q}$ is compatible, then $\mathcal{F}$ should be laminarizable.

Full Laminarization
Given: A weak $\mathcal{A}$-cut family $\mathcal{F}$.
Problem: Determine whether $\mathcal{F}$ is laminarizable or not. If $\mathcal{F}$ is laminarizable, obtain a laminar weak $\mathcal{A}$-cut family $\mathcal{L}$ with $\mathcal{L} \approx \mathcal{F}$.
Here, in FULL LAMINARIZATION, we assume that no distinct X, Y with $X \approx Y$ are contained in $\mathcal{F}$, i.e., $|\mathcal{F}| = |\mathcal{F}/{\approx}|$.

FULL LAMINARIZATION can be solved in $O(n^4)$ time by reducing to LAMINARIZATION.

▶ **Theorem 3.1.** FULL LAMINARIZATION *can be solved in $O(n^4)$ time.*

In Section 3.2, we give an $O(rn^4)$-time algorithm for FULL DISPLAYING (Theorem 3.3). Thus, by Theorems 3.1 and 3.3, we obtain Theorem 1.1 for full $\mathcal{A}$-partite quartet systems.

3.2 Algorithm for full multipartite quartet system

Our proposed algorithm for full multipartite quartet systems is devised by combining Algorithm 4 for complete multipartite quartet systems and an algorithm for full quartet systems. For full quartet system $\mathcal{Q}$, it is known [2] that QUARTET COMPATIBILITY can be solved in linear time of $|\mathcal{Q}|$, and that a phylogenetic tree displaying $\mathcal{Q}$ is uniquely determined. By summarizing these facts with notations introduced in this paper, we obtain the following.

▶ **Theorem 3.2** ([2, 8]). *Suppose that $\mathcal{Q}$ is full on $[n]$. Then* QUARTET COMPATIBILITY *can be solved in $O(|\mathcal{Q}|)$ time. Furthermore, if $\mathcal{Q}$ is compatible, then a weak $\{[n]\}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ is uniquely determined up to $\approx$.*

Let $\mathcal{A} := \{A_1, A_2, \ldots, A_r\}$ be a partition of $[n]$ with $|A_i| \geq 2$ for all $i \in [r]$. Suppose that a full $\mathcal{A}$-partite quartet system $\mathcal{Q} = \mathcal{Q}_0 \cup \mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_r$ is compatible. Then we can obtain a minimal laminarizable $\mathcal{A}$-cut family $\mathcal{F}_0$ displaying $\mathcal{Q}_0$ and laminar weak $\mathcal{A}$-cut families $\mathcal{L}_i \subseteq 2^{A_i}$ displaying $\mathcal{Q}_i$ for $i \in [r]$. By combining $\mathcal{F}_0, \mathcal{L}_1, \ldots, \mathcal{L}_r$ appropriately, we can construct a minimal laminarizable weak $\mathcal{A}$-cut family displaying $\mathcal{Q}$ as follows.

Algorithm 5 (for Full Displaying).

Input: A full $\mathcal{A}$-partite quartet system $\mathcal{Q} = \mathcal{Q}_0 \cup \mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_r$.

Output: Either detect the incompatibility of $\mathcal{Q}$, or obtain weak $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$.

Step 1: Solve Displaying for $\mathcal{Q}_0$ by Algorithm 4 and Quartet Compatibility for $\mathcal{Q}_i$ for $i \in [r]$. If algorithms detect the incompatibility of $\mathcal{Q}_i$ for some i, then output "$\mathcal{Q}$ is not compatible" and stop. Otherwise, obtain an $\mathcal{A}$-cut family $\mathcal{F}_0$ displaying $\mathcal{Q}_0$ and laminar weak $\mathcal{A}$-cut families $\mathcal{L}_i \subseteq 2^{A_i}$ displaying $\mathcal{Q}_i$ for all $i \in [r]$.

Step 2: Let $\mathcal{F}_i := \{X \cap A_i \mid X \in \mathcal{F}_0 \text{ such that } \langle X \rangle \supseteq A_i\}$ for $i \in [r]$. If $\mathcal{F}_i/{\approx} \nsubseteq \mathcal{L}_i/{\approx}$, then output "$\mathcal{Q}$ is not compatible" and stop.

Step 3: Define $\mathcal{F} := \mathcal{F}_0 \cup \bigcup_{i \in [r]} \{Y \in \mathcal{L}_i \mid Y \not\approx X \text{ for all } X \in \mathcal{F}_i\}$. If $|\mathcal{F}| \leq 2n$, then output $\mathcal{F}$. Otherwise, output "$\mathcal{Q}$ is not compatible."

▶ **Theorem 3.3.** *Algorithm 5 solves* Full Displaying *in $O(rn^4)$ time. Furthermore, if the input is compatible, then the output is a minimal laminarizable weak $\mathcal{A}$-cut family.*

By the proof of Theorem 3.3, the following corollary holds.

▶ **Corollary 3.4.** *Suppose that a full $\mathcal{A}$-partite quartet system $\mathcal{Q}$ is compatible. Then a minimal laminarizable weak $\mathcal{A}$-cut family $\mathcal{F}$ displaying $\mathcal{Q}$ is uniquely determined up to $\approx$.*

References

1. A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. 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.
2. H.-J. Bandelt and A. Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*, 7:309–343, 1986.
3. V. Berry, D. Bryant, T. Jiang, P. Kearney, M. Li, T. Wareham, and H. Zhang. A practical algorithm for recovering the best supported edges of an evolutionary tree. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms (SODA'00)*, pages 287–296, 2000.
4. V. Berry, T. Jiang, P. Kearney, M. Li, and T. Wareham. Quartet cleaning: Improved algorithms and simulations. In *Proceedings of the 7th European Symposium on Algorithm (ESA'99)*, volume 1643 of *Lecture Notes in Computer Science*, pages 313–324, Heidelberg, 1999. Springer.
5. D. Bryant and M. Steel. Extension operations on sets of leaf-labelled trees. *Advances in Applied Mathematics*, 16:425–453, 1995.
6. P. Buneman. The recovery of trees from measures of dissimilarity. In F. R. Hodson, D. G. Kendall, and P. Tautu, editors, *Mathematics in the Archaeological and Historical Science*, pages 387–395. Edinburgh University Press, 1971.
7. M.-S Chang, C.-C Lin, and P. Rossmanith. New fixed-parameter algorithms for the minimum quartet inconsistency problem. *Theory of Computing Systems*, 47(2):342–367, 2010.
8. H. Colonius and H. H. Schulze. Tree structure from proximity data. *British Journal of Mathematical and Statistical Psychology*, 34:167–180, 1981.
9. W. M. Fitch. A non-sequential method for constructing trees and hierarchical classifications. *Journal of Molecular Evolution*, 18:30–37, 1981.
10. J. Gramm and R. Niedermeier. A fixed-parameter algorithm for minimum quartet inconsistency. *Journal of Computer and System Sciences*, 67:723–741, 2003.
11. H. Hirai, Y. Iwamasa, K. Murota, and S. Živný. A tractable class of binary VCSPs via M-convex intersection. arXiv, 2018. `arXiv:1801.02199v1`.

12 T. Jiang, P. Kearney, and M. Li. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM Journal on Computing*, 30(6):1942–1961, 2001.

13 R. Reaz, M. S. Bayzid, and M. S. Rahman. Accurate phylogenetic tree reconstruction from quartets: A heuristic approach. *PLoS ONE*, 9(8):e104008, 2014.

14 S. Sattath and A. Tversky. Additive similarity trees. *Psychometrika*, 42(319–345), 1977.

15 A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Heidelberg, 2003.

16 C. Semple and M. Steel. A supertree method for rooted trees. *Discrete Applied Mathematics*, 105:147–158, 2000.

17 C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, Oxford, 2003.

18 M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.

19 K. Strimmer and A. Haeseler. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Journal of Molecular Biology and Evolution*, 13:964–969, 1996.