On the maximum quartet distance between phylogenetic trees

Noga Alon [*] Humberto Naves [†] Benny Sudakov [‡]

Abstract

A conjecture of Bandelt and Dress states that the maximum quartet distance between any two phylogenetic trees on n leaves is at most $(\frac{2}{3}+o(1))\binom{n}{4}$. Using the machinery of flag algebras we improve the currently known bounds regarding this conjecture, in particular we show that the maximum is at most $(0.69+o(1))\binom{n}{4}$. We also give further evidence that the conjecture is true by proving that the maximum distance between caterpillar trees is at most $(\frac{2}{3}+o(1))\binom{n}{4}$.

1 Introduction

The practice of phylogenetic tree reconstruction to hypothesize various aspects of evolutionary relationships among different species of organisms has become a central problem in molecular biology. For instance, the "Tree of Life" project [15] aims, among other things, to accurately construct a tree representing the evolutionary history of the organismal lineages as they change through time.

A phylogeny (the evolutionary history of a set of species) is usually represented by a tree where the species under study are mapped to the leaves of the tree and the tree-structure represents the different evolutionary relationships among them. Here we focus solely on *undirected* (or *unrooted*) phylogenetic trees. In this setting, the underlying tree is not directed and each non-leaf node is incident to exactly three edges. The basic unit of information for phylogenetic classification is the *quartet*, which is an undirected phylogenetic tree having exactly four leaves. We denote a quartet over the leaves $\{a,b,c,d\}$ as $[ab|cd]$ whenever there is an edge in the underlying tree separating the pair $\{a,b\}$ from the pair $\{c,d\}$, as Figure 1 shows. Note that a phylogenetic tree defined over a taxa (species) set of size n contains the information of exactly $\binom{n}{4}$ quartets.

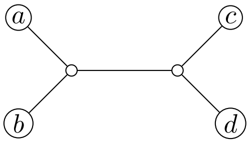

Figure 1: A quartet.

Studying quartets is of prime importance not only because they are the smallest informational units induced by a phylogeny, but also because they play a major role in many reconstruction methods. Among them, the *quartet-based reconstruction* is perhaps the most basic and most studied approach (see e.g. [5, 6, 12, 13, 22, 23, 25]). The task of the quartet-based reconstruction is to find a tree over the full set of species that satisfies most of the given input quartets. In its full generality this problem is very difficult as Steel [24] has shown that even deciding if there is a tree that satisfies all the input quartets is NP-complete. To aggravate matters, even the ideal case in which all quartets agree on a single tree is very rare. Thus a natural problem arises, namely, finding a tree maximizing the number of compatible quartets — *maximum quartet compatibility* (MQC) [21]. As MQC is obviously NP-hard, several approximation algorithms have been sugested. However, the best known approximation to the general problem is still obtained by a naive "random labelling of the leaves of a tree" with expected approximation ratio of 1/3.

Related to the problem of compatibility is the concept of *quartet distance* [9]. This notion is used to measure similarity of two different phylogenetic trees by means of counting how many quartets are compatible to both of them. More specifically, if T_1 and T_2 are two phylogenetic trees on n leaves, let $qd(T_1,T_2)$ denote the difference between $\binom{n}{4}$ and the number of quartets compatible to both T_1 and T_2. With this definition in mind, a natural question emerges: what

[*] Sackler School of Mathematics and Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel and School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540. Email: nogaa@tau.ac.il. Research supported in part by a USA-Israeli BSF grant, by an ISF grant, by the Israeli I-Core program and by the Fund for Mathematics.

[†] Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN 55455, USA. Email: hnaves@ima.umn.edu. This research was supported in part by the Institute for Mathematics and its Applications with funds provided by the National Science Foundation.

[‡] Department of Mathematics, ETH, 8092 Zurich. Email: benjamin.sudakov@math.ethz.ch. Research supported in part by SNSF grant 200021-149111 and by a USA-Israeli BSF grant.

is the maximum quartet distance between two phylogenetic trees on n leaves? Somewhat surprisingly the answer is strictly smaller than $\binom{n}{4}$. Bandelt and Dress [4] showed that the maximum is always strictly smaller than $\frac{14}{15}\binom{n}{4}$ for $n \geq 6$. They also conjectured that the ratio between the maximum quartet distance and $\binom{n}{4}$ converges to $\frac{2}{3}$ as n tends to infinity.

Conjecture 1.1. (Bandelt and Dress) *The maximum quartet distance between two phylogenetic trees on n leaves is $\left(\frac{2}{3} + o(1)\right)\binom{n}{4}$.*

Alon, Snir, and Yuster [1] further improved the bounds on the maximum quartet distance. Namely, they proved that the maximum is strictly larger than $\frac{2}{3}\binom{n}{4}$ for every n but asymptotically smaller than $\frac{9}{10}\binom{n}{4}$. The lower bound of $\frac{2}{3}\binom{n}{4}$ can be again obtained by the same "random labelling of the leaves" argument, thus Conjecture 1.1 implies that the average distance between two random trees is asymptotically the same as the maximum distance. We also remark that the problem of maximizing the quartet-distance between trees can be rephrased as how much a compatible set of quartets can be violated, which is the opposite of MQC.

The main contribution of this paper is the following statement, which we obtain using the machinery of flag algebras developed by Razborov in [18].

Theorem 1.1. *The maximum quartet distance between two phylogenetic trees on n leaves is at most $(0.69 + o(1))\binom{n}{4}$.*

As further evidence that $\frac{2}{3}\binom{n}{4}$ is the correct answer, we prove the following statement which establishes Conjecture 1.1 when restricted to *caterpillar trees*. By *caterpillar* we mean a phylogenetic tree having at most two vertices which are each adjacent to two leaves, as in Figure 2.

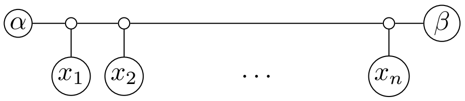

Figure 2: A caterpillar with $n + 2$ leaves.

Theorem 1.2. *The maximum quartet distance between two phylogenetic caterpillar trees on n leaves is at most $\left(\frac{2}{3} + o(1)\right)\binom{n}{4}$.*

The set of all caterpillar trees is a simple yet very important subclass of phylogenetic trees. For instance, the proof of NP-hardness of MQC by Steel [24] heavily relies on this particular subclass. Namely, deciding if there exists a tree T that satisfies all the quartets in a given input set is NP-complete even if we further assume that T is caterpillar.

In Section 5, we show that the problem of computing the maximum quartet distance between two caterpillar trees can be essentially reduced to the problem of computing the density of the following induced sub-permutations in a permutation $\pi \in S_n$:

$$1234, 1243, 2134, 2143, 3412, 4312, 3421, 4321.$$

The rest of this paper is organized as follows. In Section 2, we formally define all the relevant notions in phylogenetics that were briefly mentioned in this introduction. In Section 3, we provide an informal explanation of our main tool, flag algebras. In Section 4, we discuss some of the details of the proof of Theorem 1.1 and provide a link to the program establishing the proof. In addition, we give the proof of Theorem 1.2 in Section 5. Lastly, the final section contains some concluding remarks and open problems.

2 Preliminaries

A *trivalent tree* is a tree in which all internal vertices (the non-leaves) have exactly three neighbors. Whenever the leaves of such trees are labeled bijectively by a taxa (species) set $\mathcal{X}$ of size n, we shall call them *phylogenetic trees*. Throughout this paper, unless stated otherwise, all trees are assumed to be phylogenetic trees. For a tree $T = (V, E)$, the set of leaves of T is denoted by $\mathcal{L}(T)$.

The removal of an edge e in a phylogenetic tree splits it into two subtrees, and thus induces a *split* among the leaves of the tree. We identify an edge e by the split $(U, \mathcal{L}(T) \setminus U)$ it generates on the set of leaves, and denote this split by U_e. As external edges (the ones adjacent to the leaves) induce trivial splits, we consider only the ones induced by internal edges.

Let T be a tree and $A \subseteq \mathcal{L}(T)$ a subset of the leaves of T. We denote by $T|_A$, the topological subtree of T induced by A were all leaves in $\mathcal{L}(T) \setminus A$ and paths leading exclusively to them are removed, and subsequently internal vertices with degree two are contracted.

For two trees T and T', we say that T *satisfies* T' (or , equivalently, that T' is *satisfied* by T), if $\mathcal{L}(T') \subseteq \mathcal{L}(T)$ and $T|_{\mathcal{L}(T')} \simeq T'$, that is, the subgtree of T induced by $\mathcal{L}(T')$ is isomorphic to T'. Otherwise, T' is *violated* by T. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of trees with possibly overlapping leaves, and denote by $\mathcal{L}(\mathcal{T}) = \bigcup_i \mathcal{L}(T_i)$, the union of the set of leaves of all trees $T_i \in \mathcal{T}$. Then for a tree T with $\mathcal{L}(T) = \mathcal{L}(\mathcal{T})$, we denote by $\mathcal{T}_s(T)$ the set of trees in $\mathcal{T}$ that are satisfied by T. We say that $\mathcal{T}$ is *compatible* if there exists a tree T^* over the set of leaves $\mathcal{L}(\mathcal{T})$ that satisfies every tree $T_i \in \mathcal{T}$, i.e. $\mathcal{T}_s(T^*) = \mathcal{T}$ (see Figure 3). We denote by $co(\mathcal{T})$ the set of trees that satisfy $\mathcal{T}$ (up to isomorphisms), $co(\mathcal{T}) = \{T : \mathcal{T}_s(T) = \mathcal{T}\}$.

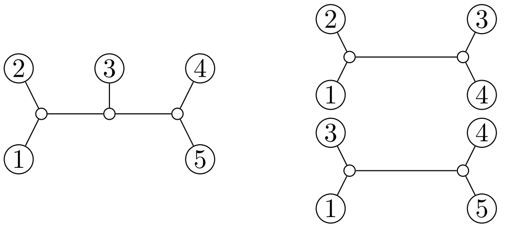

Figure 3: A tree on five leaves and two quartets compatible with it.

Further, we say that T^* is *defined* by $\mathcal{T}$ if $co(\mathcal{T})$ is the singleton $\{T^*\}$. If there is no such compatible tree T^*, we say that $\mathcal{T}$ is *incompatible* (i.e., $co(\mathcal{T}) = \emptyset$).

A *quartet* tree (or just a quartet for short), is a phylogenetic tree over four leaves $\{a, b, c, d\}$. We denote a quartet over $\{a, b, c, d\}$ as $[ab|cd]$ if there exists an edge e whose corresponding split U_e satisfies $a, b \in U$ and $c, d \notin U$. Quartets are the most elementary informational unit in a phylogenetic tree, as a pair corresponds to a path in a tree and a triplet to a vertex (the unique vertex in the intersection of all the pairwise paths connecting the three leaves). Every phylogenetic tree T with n leaves defines $\binom{n}{4}$ quartets, one for each set of four leaves. Let $\mathcal{Q}(T)$ denote this full quartet set of T. It is well-known that $\mathcal{Q}(T)$ uniquely defines T. In fact Colonius and Schulze [7] showed that the following proposition holds.

Proposition 2.1. (Colonius and Schulze) *Let $\mathcal{Q}$ be a full quartet set over n species. If every subset of three quartets (a quartet triplet) is compatible, then $\mathcal{Q}$ is compatible and there exists a unique tree defined by $\mathcal{Q}$. In fact, if for every five taxa $\{a, b, c, d, e\}$ the following holds:*

- *if $\{[ae|bc], [ae|cd]\} \cap \mathcal{Q} \neq \emptyset$ and $[ab|cd] \in \mathcal{Q}$ then $[be|cd] \in \mathcal{Q}$,*

then $\mathcal{Q}$ is compatible.

Lastly, we would like to briefly sketch the "random labelling of the leaves" argument. Let T be any tree with n leaves labeled by a taxa set $\mathcal{X}$. Consider a random bijection π between $\mathcal{X}$ and the leaves of T. The corresponding labeled tree is denoted by T^π. As each of the $n!$ possible bijections is equally likely, we notice that a quartet $[ab|cd]$ with labels from $\mathcal{X}$ is satisfied by T^π with probability exactly 1/3. Thus, by linearity of expectation, we have:

Proposition 2.2. *Let $\mathcal{Q}$ be an arbitrary set of quartets over a taxa set $\mathcal{X}$ of size n, and let T^π be a random bijection between the leaves of a tree T and $\mathcal{X}$. The expected number of elements in $\mathcal{Q}$ satisfied by T is $|\mathcal{Q}|/3$.*

As a consequence, we have the next statement.

Proposition 2.3. *Let T_1 and T_2 be two random phylogenetic trees over the same taxa set $\mathcal{X}$ of size n, sampled independently and uniformly at random. The expected value of the quartet distance $qd(T_1, T_2)$ is $\frac{2}{3}\binom{n}{4}$.*

3 Flag algebra calculus

In this section we provide a brief introduction to the technique of flag algebras. First introduced by Razborov in [18], it has been applied with great success to a wide variety of problems in extremal combinatorics (see, for example, [2, 3, 8, 10, 11, 16, 17, 19, 20] and many others).

We begin with a brief explanation on how to map the problem of finding the maximum quartet distance into a problem in extremal combinatorics. We then proceed with a general overview of the flag algebra calculus in the second subsection, by introducing some key definitions and providing some intuition behind the machinery. The third subsection will show how we express extremal problems in the language of flag algebras. It is neither our goal to be rigorous nor thorough, but rather to emphasize that the combinatorial arguments behind the flag algebra calculus are as old as extremal combinatorics itself. Indeed, the main tools available to us are double-counting and the Cauchy-Schwarz inequality.

The flag algebra calculus is powerful because it provides a formalism through which the combinatorial problem can be reduced to a semi-definite programming (SDP) problem. This in turn enables the use of computers to find solutions, with rigorous proofs, to problems in extremal combinatorics. For a more complete survey of the technique, we refer the reader to the excellent expositions in [14] and [17], while for a technical specification of flag algebras, we suggest the original paper of Razborov [18].

3.1 The model In this section, the main object of interest is the *tree-pair*. From two phylogenetic trees T_1 and T_2 labelled by the same taxa set, we would like to create a simple object that "represents" the pair (T_1, T_2) in such a way that we can still compute the quartet distance $qd(T_1, T_2)$ from it. Note that the actual set of labels (the taxa set) is irrelevant in the computation of this distance, so this object shall have no labels at all. A natural and amenable definition comes to mind. A *tree-pair* D is a pair of trivalent trees $D = (\overline{T}_1, \overline{T}_2)$ (i.e., unlabelled phylogenetic trees) having the same set of leaves but having no other vertex in common. In that case we write $\mathcal{L}(D) := \mathcal{L}(\overline{T}_1) = \mathcal{L}(\overline{T}_2)$. From two phylogenetic trees T_1 and T_2 over the same taxa set, one can construct a tree-pair D in the following way. We first identify leaves from T_1 and T_2 having the same label and we subsequently remove labels from T_1 and T_2 altogether to obtain $\overline{T}_1$ and $\overline{T}_2$, respectively. We often represent a tree-pair $D = (\overline{T}_1, \overline{T}_2)$ by the graph $\overline{T}_1 \cup \overline{T}_2$ which is the *union* of $\overline{T}_1$ and $\overline{T}_2$, that is, $V(\overline{T}_1 \cup \overline{T}_2) = V(\overline{T}_1) \cup V(\overline{T}_2)$ and $E(\overline{T}_1 \cup \overline{T}_2) = E(\overline{T}_1) \cup E(\overline{T}_2)$, with $\overline{T}_1$ positioned "on top" of $\overline{T}_2$. (see Figure 4).

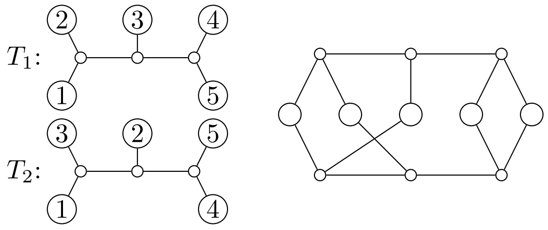

Figure 4: Two trees over the same taxa set and the tree-pair formed by their union.

Two tree-pairs $D = (\overline{T}_1, \overline{T}_2)$ and $D' = (\overline{T}'_1, \overline{T}'_2)$ are *isomorphic* if there exists an isomorphism between the graphs $\overline{T}_1 \cup \overline{T}_2$ and $\overline{T}'_1 \cup \overline{T}'_2$ that maps vertices of $\overline{T}_i$ to vertices of $\overline{T}'_i$ for $i = 1, 2$, i.e., it respects the component trees. To indicate that two tree-pairs are isomorphic we write $D \simeq D'$. For a tree-pair $D = (T'_1, T'_2)$ and a subset $A \subseteq \mathcal{L}(D)$, we write $D|_A$ to denote the *sub-tree-pair induced by* A, that is, $D|_A := (T'_1|_A, T'_2|_A)$. Recall that to obtain $T'_1|_A$ and $T'_2|_A$ we keep the smallest subtrees of T'_1 and T'_2 (respectively) containing the leaves in A. We subsequently delete one by one all vertices of degree two by replacing the corresponding path of length 2 by an edge until all the vertices not in A have degree 3. For two tree-pairs D and D', if there exists a set $A \subseteq \mathcal{L}(D)$ such that $D|_A \simeq D'$, then we say that D contains an *induced copy* of D'.

There are only 2 non-isomorphic tree-pairs having exactly four leaves, namely id_4 and cr_4 (see Figure 5).

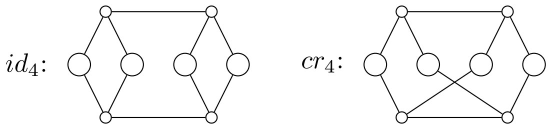

Figure 5: The two non-isomorphic tree-pairs on four leaves.

A moments thought reveals that the quartet distance $qd(T_1, T_2)$ can be computed as follows. Let $D = (\overline{T}_1, \overline{T}_2)$ be the tree-pair obtained by joining T_1 and T_2. Then $qd(T_1, T_2)$ is simply the number of induced copies of cr_4 in D. Hence to prove Theorem 1.1, it suffices to show that the following is true.

Theorem 3.1. *Let D be any tree-pair on n leaves. The number of induced copies of cr_4 in D is at most* $(0.69 + o(1))\binom{n}{4}$.

3.2 Definitions and notation in the calculus

The flag algebra calculus is typically used to find the extremal density of some fixed structure in a given family of combinatorial objects. In our case (see Theorem 3.1), it will be used to maximize the density of cr_4 among all tree-pairs of size n, for n sufficiently large. While the theory of flag algebras is very general and can be applied to several different types of problems, we will explain it using only examples related to our particular setting.

A *type* σ is a labeled tree-pair using labels from $[k]$, where $k = |\mathcal{L}(\sigma)|$. That is, each leaf in $\mathcal{L}(\sigma)$ is associated with a label from $[k]$, where k is a nonnegative integer. The *size* of σ is the integer k, and is denoted by $|\sigma|$. Figure 6 shows some examples of types.

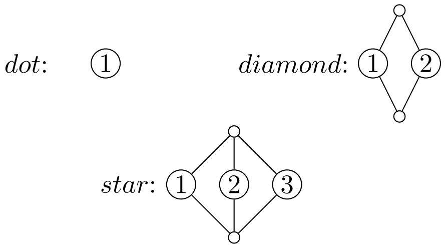

Figure 6: Examples of types.

In what follows, an isomorphism between tree-pairs must preserve any labels that are present. Given a type σ, a *σ-flag* is a tree-pair F on a partially labeled set of leaves, such that the sub-tree-pair induced by the labeled leaves is isomorphic to σ. The *underlying tree-pair* of the flag F is the tree-pair F with all labels removed. The *size* of a flag is the number of leaves, that is, $|\mathcal{L}(F)|$. Note that when σ is the *trivial type* of size 0 (denoted by $\sigma = 0$), a σ-flag is just a usual unlabeled tree-pair. We shall write $\mathcal{F}_l^\sigma$ for the collection of all σ-flags of size l (up to isomorphism). In Figure 7 we list all flags in $\mathcal{F}_4^{dot}$. Let $\mathcal{F}^\sigma = \bigcup_{l \geq |\sigma|} \mathcal{F}_l^\sigma$. When the type σ is trivial, we shall omit the superscript from our notation.

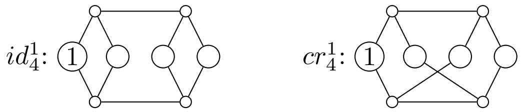

Figure 7: Family $\mathcal{F}_4^{dot}$.

Let us now define two fundamental concepts in our calculus, namely those of flag densities in larger flags and tree-pairs. Let σ be a type of size k, let $m \geq 1$ be an integer and let $\{F_i\}_{i=1}^m$ be a collection of σ-flags of sizes $l_i = |F_i| \geq k$. Given a σ-flag F of order at least $l = k + \sum_{i=1}^m (l_i - k)$, let $A \subseteq \mathcal{L}(F)$ be the set of labeled leaves of F. Now select disjoint subsets $X_i \subseteq \mathcal{L}(F) \setminus A$ of sizes $|X_i| = l_i - k$, uniformly at random. This is possible because F has at least $\sum_i (l_i - k)$ unlabeled leaves. Denote by E_i the event that the σ-flag induced by $A \cup X_i$ is isomorphic to F_i, for $i \in [m]$. We define $p_\sigma(F_1, F_2, \ldots, F_m; F) := \mathbb{P}\left[\bigcap_{i=1}^m E_i\right]$ to be the probability that all these events occur simultaneously.

If D is just a tree-pair of order at least l, and not a σ-flag, then there is no pre-labeled set of leaves A that induces the type σ. Instead, we uniformly at random select a partial labeling $L : [k] \to \mathcal{L}(D)$. This random labeling turns D into a σ'-flag F_L, where the type σ' is the labeled sub-tree-pair induced by the set of vertices $L([k])$. If $\sigma' = \sigma$, we can then proceed as above, otherwise we say the events E_i have probability 0. Finally, we average over all possible random labellings. Formally, let Y be the random variable defined by $Y = \mathbf{1}_{[\sigma' = \sigma]} \cdot p_\sigma(F_1, F_2, \ldots, F_m; F_L)$, where $\mathbf{1}_{[\sigma' = \sigma]}$ is the indicator of the event $\sigma' = \sigma$. Define $d_\sigma(F_1, \ldots, F_m; D) = \mathbb{E}[Y]$ as the expected value of the random variable Y. The quantities $p_\sigma(F_1, F_2, \ldots, F_m; F)$ and $d_\sigma(F_1, F_2, \ldots, F_m; D)$ are called *flag densities* of $\{F_i\}_{i \in [m]}$ in F and in D, respectively. Clearly these flag densities are the same whenever $\sigma = 0$, in which case we omit the subscript from both notations.

To better illustrate these definitions, we give some examples. These flags are shown in Figure 8.

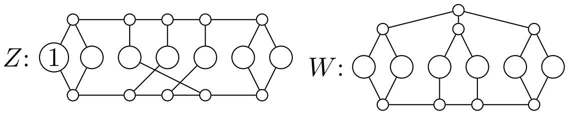

Figure 8: Example flags.

We turn to compute the flag densities of id_4^1 and cr_4^1 in the flag Z. For example, to compute $p_{dot}(id_4^1; Z)$, note that to induce a copy of id_4^1 we must choose exactly 3 other unlabeled leaves which together with the labeled leaf 1 induce a copy of id_4^1. There are $\binom{6}{3} = 20$ ways to make the choice of 3 unlabeled leaves, and out of the 20 exactly 15 induce a copy of id_4^1, thus $p_{dot}(id_4^1; Z) = \frac{3}{4}$. Similarly we obtain $p_{dot}(cr_4^1; Z) = \frac{1}{4}$. We can also compute the joint flag densities of multiple flags. For instance, let us consider $p_{dot}(id_4^1, id_4^1; Z)$. In this case, we first choose a set X_1 of 3 unlabeled leaves uniformly at random and we subsequently choose a set X_2 of 3 other unlabeled leaves also uniformly at random. Since the choice of X_2 is uniquely determined given the choice of X_1, there are exactly $\binom{6}{3}$ possible choices for the pair (X_1, X_2). Out of these 20 choices, one can count that exactly 10 of them will be such that both X_1 and X_2 will induce a copy of id_4^1 when we add the labeled leaf. The order matters here: when computing $p(F_1, F_2; F)$, the set X_1 must induce a copy of F_1 while X_2 must induce a copy of F_2. Thus $p_{dot}(id_4^1, id_4^1; Z) = \frac{1}{2}$. Similarly, we have $p_{dot}(id_4^1, cr_4^1; Z) = p_{dot}(cr_4^1, id_4^1; Z) = \frac{1}{4}$ and $p_{dot}(cr_4^1, cr_4^1; Z) = 0$.

The computation of flag densities d_{dot} for unlabeled tree-pairs is a little more involved. To see how to compute it, we consider W depicted in Figure 8 as an example. There are two non-isomorphic *dot*-flags whose underlying tree-pair is W, namely W_1^1 and W_2^1 as shown in Figure 9.

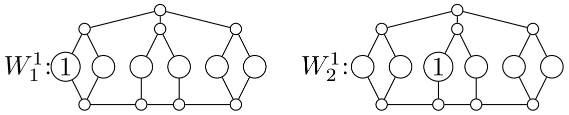

Figure 9: *dot*-flags for W.

If we randomly label a leaf from W, then with probability $\frac{2}{3}$ it will become W_1^1 and with probability $\frac{1}{3}$ it will become W_2^1. Moreover, since $p(id_4^1; W_1^1) = \frac{4}{5}$ and $p_{dot}(id_4^1; W_2^1) = \frac{3}{5}$, we have $d_{dot}(id_4^1; W) = \frac{2}{3} p_{dot}(id_4^1; W_1^1) + \frac{1}{3} p_{dot}(id_4^1; W_2^1) = \frac{11}{15}$. Similarly we have $p_{dot}(cr_4^1; W_1^1) = \frac{1}{5}$ and $p_{dot}(cr_4^1; W_2^1) = \frac{2}{5}$, hence $d_{dot}(cr_4^1; W) = \frac{4}{15}$.

The reader might notice that there is an alternative way to compute, say, $d_{dot}(cr_4^1; W)$: we simply compute the product of $d_{dot}(cr_4^1; cr_4) \cdot p(cr_4; W) =$

$1 \cdot \frac{4}{15} = \frac{4}{15}$. In general, suppose as before that we have a type σ of size k, a σ-flag F of size $l \geq k$, and an unlabeled tree-pair D. To compute $d_\sigma(F; D)$, we averaged over all random partial labelings of D the probability of finding a flag isomorphic to F. A simple double-counting argument shows that we can do the "averaging" before the random labeling, which is the idea behind Razborov's *averaging operator*, as defined in Section 2.2 of [18]. Let $F|_0$ denote the unlabeled underlying model of F. We can compute $d_\sigma(F; D)$ by first computing $d(F|_0; D)$, the probability that l randomly chosen vertices in D form an induced copy of $F|_0$ as a sub-model. Given this copy of $F|_0$, we then randomly label k of the l vertices, and compute the probability that these k vertices are label-isomorphic to σ. This amounts to multiplying $d(F|_0; D)$ by a *normalizing factor* $q_\sigma(F)$, that is, $d_\sigma(F; D) = q_\sigma(F)d(F|_0; D) = q_\sigma(F)p(F|_0; D)$. We can interpret the normalizing factor as $q_\sigma(F) = d_\sigma(F; F|_0)$.

There are more relations involving d_σ and p_σ than the one mentioned previously. We will now state, without proof, a basic fact about flag densities that can be proved easily by double-counting.

PROPOSITION 3.1. (CHAIN RULE) *If σ is a type of size k, $m \geq 1$ is an integer, and $\{F_i\}_{i=1}^m$ is a family of σ-flags of sizes $|F_i| = l_i$, and $l \geq k + \sum_{i=1}^m (l_i - k)$ is an integer parameter, then*

1. *For any σ-flag F of order at least l, we have $p_\sigma(F_1, \dots, F_m; F)$ equals to*

$$\sum_{F' \in \mathcal{F}_l^\sigma} p_\sigma(F_1, \dots, F_m; F')p_\sigma(F'; F).$$

2. *For any tree-pair D of size at least l, we have $d_\sigma(F_1, \dots, F_m; D)$ equals to*

$$\sum_{H \in \mathcal{F}_l} d_\sigma(F_1, \dots, F_m; H)d(H; D),$$

which is also equal to

$$\sum_{F \in \mathcal{F}_l^\sigma} p_\sigma(F_1, \dots, F_m; F)d_\sigma(F; D).$$

To illustrate the chain rule for $m = 1$ and $\sigma = 0$, we consider the "expansion" of id_4 in $\mathcal{F}_5$ (see Figure 10).

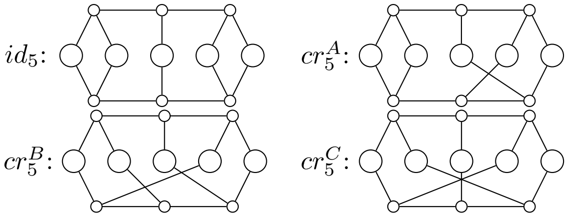

Figure 10: Family $\mathcal{F}_5$.

The chain rule gives

$$\begin{aligned} p(id_4; F) &= p(id_4; id_5)p(id_5; F) \\ &\quad + p(id_4; cr_5^A)p(cr_5^A; F) \\ &\quad + p(id_4; cr_5^B)p(cr_5^B; F) \\ &\quad + p(id_5; cr_5^C)p(cr_5^C; F) \\ &\quad = p(id_5; F) + \frac{3}{5}p(cr_5^A; F) + \frac{1}{5}p(cr_5^B; F). \end{aligned}$$

Similarly, we can expand $p(cr_4; F) = \frac{2}{5}p(cr_5^A; F) + \frac{4}{5}p(cr_5^B; F) + p(cr_5^C; F)$. For the ease of notation, we can express these two identities using the syntax of flag algebras:

$$\begin{aligned} id_4 &= id_5 + \frac{3}{5}cr_5^A + \frac{1}{5}cr_5^B \\ cr_4 &= \frac{2}{5}cr_5^A + \frac{4}{5}cr_5^B + cr_5^C. \end{aligned}$$

In this syntax, the equation $\sum_{i \in I} \alpha_i F_i = 0$ means that for all sufficiently large σ-flags F, we have $\sum_{i \in I} \alpha_i p_\sigma(F_i; F) = 0$, where $\alpha_i \in \mathbb{R}$ for all $i \in I$. We use $\mathcal{A}^\sigma$ to denote the set of linear combinations of flags of type σ. It is convenient to define a *product* of flags in the following way:

$$F_1 \cdot F_2 := \sum_{F \in \mathcal{F}_l^\sigma} p_\sigma(F_1, F_2; F)F,$$

where $F_1 \in \mathcal{F}^\sigma$, $F_2 \in \mathcal{F}^\sigma$, $l \geq |F_1| + |F_2| - |\sigma|$. (Note that because of the chain rule, it does not matter which l we choose.)

To further simplify the notation, we can extend the definitions of p_σ and d_σ to $\mathcal{A}^\sigma$ by making them linear in each coordinate. The product notation simplifies these extended definitions, because $p_\sigma(f_1 \cdot f_2; f) = p_\sigma(f_1, f_2; f)$ and $d_\sigma(f_1 \cdot f_2; g) = d_\sigma(f_1, f_2; g)$, for any $f_1, f_2, f \in \mathcal{A}^\sigma$ and for any $g \in \mathcal{A}^0$.

The last piece of notation we introduce is that of the averaging operator. Recall that for any σ-flag F, we had the normalizing factors $q_\sigma(F)$ such that $d_\sigma(F; G) = q_\sigma(F)p(F|_0; G)$. In the syntax of flag algebra, this averaging operation is denoted by $[[F]]_\sigma := q_\sigma(F) \cdot F|_0$. We extend this linearly to all elements of $\mathcal{A}^\sigma$. For example

$$[[id_4^1]]_{dot} = id_4, \quad [[cr_4^1]]_{dot} = cr_4,$$

$$[[id_4^1 + cr_4^1]]_{dot} = id_4 + cr_4, \text{ and } [[W_1^1]]_{dot} = \frac{2}{3}W.$$

This notation is useful, because $d_\sigma(f; g) = p([[f]]_\sigma; g)$ for any $f \in \mathcal{A}^\sigma$ and for any $g \in \mathcal{A}^0$, and hence we have a unified notation for both types of flag densities.

3.3 Extremal problems in the flag algebra calculus Recall that our optimization problem is to maximize the density of cr_4 amongst all possible tree-pairs. We will show how flag algebras can be applied to this problem to reduce it to a semi-definite programming (SDP) problem, which can then be solved numerically.

We may use the chain rule to obtain $d(cr_4; D) = \sum_{H\in\mathcal{F}_t} d(cr_4; H)d(H; D)$ for $t \geq 4$. Since $\sum_{H\in\mathcal{F}_t} d(H; D) = 1$, we have $d(cr_4; D) \leq \max_{H\in\mathcal{F}_t} d(cr_4; H)$, which is a bound that clearly does not depend on D. For instance, when we choose $t = 6$ we already obtain $d(cr_4; D) \leq \frac{14}{15}$.

Inequalities obtained this way are often very weak, since we only use very local considerations about the sub-tree-pairs $H \in \mathcal{F}_t$, and we do not take into account how the tree-pairs fit together in the larger tree-pair D; that is, how they intersect.

One might hope to find inequalities of the form $\sum_{H\in\mathcal{F}_t} \alpha_H d(H; D) \geq 0$, such that when we combine them with the initial identity, we get

$$\begin{aligned} d(cr_4; D) &\leq d(cr_4; D) + \sum_{H\in\mathcal{F}_t} \alpha_H d(H; D) \\ &= \sum_{H\in\mathcal{F}_t} (d(cr_4; H) + \alpha_H)d(H; D) \\ &\leq \max_{H\in\mathcal{F}_t} \{d(cr_4; H) + \alpha_H\}. \end{aligned}$$

Since α_H can be negative for some models H, the hope is that this will improve the low coefficients by transferring weight from high coefficients. In order to find such inequalities, we need another property of the flag densities.

Proposition 3.2. *If σ is a fixed type of size k, $m \geq 1$ is an integer, $\{F_i\}_{i=1}^m$ is a fixed family of σ-flags of sizes $|F_i| = l_i$, and $l \geq k + \sum_{i=1}^m (l_i - k)$ is an integer, then for any flag F of order $n \geq l$, we have*

$$p_\sigma(F_1, \ldots, F_m; F) = \left[\prod_{i=1}^m p_\sigma(F_i; F)\right] + O(1/n),$$

where the constant in the big-O notation might depend on the family $\{F_i\}_{i=1}^m$.

One can prove Proposition 3.2 by noting that, if we drop the requirement that the sets X_i are disjoint in the definition of $p_\sigma(F_1, \ldots, F_m; F)$, the events E_i will become independent, and thus $\mathbb{P}[\cap_{i=1}^m E_i] = \prod_{i=1}^m \mathbb{P}[E_i] = \prod_{i=1}^m p_\sigma(F_i; F)$. The error introduced is the probability that these sets X_i will intersect in F, which is $O(1/n)$. It is tempting to claim a similar product formula for the unlabeled flag densities d_σ, but we cannot do so. In the above equation, it is essential that all the σ-flags F_i share the same labeled type σ, and hence we require F to be a σ-flag.

We are now ready to establish some inequalities. Let's first fix a type σ of size k. If Q is any positive semi-definite $|\mathcal{F}_l^\sigma| \times |\mathcal{F}_l^\sigma|$ matrix with rows and columns indexed by the same set $\mathcal{F}_l^\sigma$, where $l \geq k$, define the "quadratic form" on flags by $Q\{\mathcal{F}_l^\sigma\} := \sum_{F_1,F_2\in\mathcal{F}_l^\sigma} Q_{F_1,F_2} F_1 \cdot F_2$. Note that $Q\{\mathcal{F}_l^\sigma\} \in \mathcal{A}^\sigma$. Proposition 3.2 yields that, for a σ-flag F of sufficiently large size, $p_\sigma(Q\{\mathcal{F}_l^\sigma\}; F)$ can be approximated as

$$\sum_{F_1,F_2\in\mathcal{F}_l^\sigma} Q_{F_1,F_2} p_\sigma(F_1; F) p_\sigma(F_2; F). \tag{3.1}$$

Note that because Q is positive semi-definite, the summation in (3.1) is always non-negative. Even after averaging we obtain:

$$\begin{aligned} [[Q]]_\sigma(D) &:= p([[Q\{\mathcal{F}_l^\sigma\}]]_\sigma; D) \\ &= \sum_{F_1,F_2\in\mathcal{F}_l^\sigma} Q_{F_1,F_2} d_\sigma(F_1, F_2; D) \\ &= \sum_{F_1,F_2\in\mathcal{F}_l^\sigma} Q_{F_1,F_2} \cdot \left(\sum_{F\in\mathcal{F}_t^\sigma} p_\sigma(F_1, F_2; F) d_\sigma(F; D)\right) \\ &= O(1/n) + \sum_{F\in\mathcal{F}_t^\sigma} \Bigg[d_\sigma(F; D) \cdot \\ &\qquad \cdot \left(\sum_{F_1,F_2\in\mathcal{F}_l^\sigma} Q_{F_1,F_2} p_\sigma(F_1; F) p_\sigma(F_2; F)\right)\Bigg] \\ &\geq o_{n\to\infty}(1), \end{aligned}$$

where n is the size of the tree-pair D and $2l - |\sigma| \leq t \leq n$ is some fixed integer. Therefore, when n is large, we have that $[[Q]]_\sigma(D)$ is asymptotically non-negative. For each admissible model H of size exactly t, let $\alpha_H = [[Q]]_\sigma(H) = \sum_{F_1,F_2\in\mathcal{F}_t^\sigma} Q_{F_1,F_2} d_\sigma(F_1, F_2; H)$. We then have

$$[[Q]]_\sigma(D) = \sum_{H\in\mathcal{F}_t} \alpha_H d(H; D) \geq o_{n\to\infty}(1).$$

The expression in the middle of the above equation is called the *expansion* of $[[Q]]_\sigma(D)$ in tree-pairs of size t, with α_H the coefficients of the expansion. For the sake of conciseness, we often omit the parameter D and express this asymptotic inequality (combined with the expansion in size t) in the syntax of flag algebras

$$\begin{aligned}[[Q_\sigma]] &:= [[Q\{\mathcal{F}_l^\sigma\}]]_\sigma \\ &= \left[\left[\sum_{F_1,F_2\in\mathcal{F}_l^\sigma} Q_{F_1,F_2} F_1\cdot F_2\right]\right]_\sigma \\ &= \sum_{H\in\mathcal{F}_t}\alpha_H H \geq 0.\end{aligned} \tag{3.2}$$

(Note that all inequalities between flags stated in the language of flag algebras are asymptotic.)

In general, if we have more than one inequality available, we can combine them together, provided they are all expanded in the same size t. Suppose we have r inequalities given by the positive semi-definite matrices Q_i of the σ_i-flags of size l_i. Adding them together, we obtain $\sum_{i=1}^r [[Q_i]]_{\sigma_i} = \sum_{H\in\mathcal{F}_t}\alpha_H H \geq 0$, where

$$\alpha_H = \sum_{i=1}^r \left(\sum_{F_1,F_2\in\mathcal{F}_{l_i}^{\sigma_i}} (Q_i)_{F_1,F_2} d_{\sigma_i}(F_1,F_2;H) \right),$$

and we want to minimize $\max_{H\in\mathcal{F}_t}\{d(cr_4;H)+\alpha_H\}$.

Thus we have transformed the original problem of finding a minimum upper bound for $d(cr_4;G)$ into a linear system involving the variables $(Q_i)_{F_k,F_l}$. As we have the constraint that the matrices Q_i should be positive semi-definite, this is a semi-definite programming problem. To take the maximum coefficient in the expansion, we introduce an artificial variable y, and require it to be bounded below by all the coefficients. Hence we have the following SDP problem in the variables y and $(Q_i)_{F_1,F_2}$:

Minimize y, subject to the constraints:

- We have $s_H \geq 0$ for all $H\in\mathcal{F}_t$, where

  $$\begin{aligned} s_H &:= y - d(cr_4;H) \\ &- \sum_{i=1}^r \left(\sum_{F_1,F_2\in\mathcal{F}_{l_i}^{\sigma_i}} (Q_i)_{F_1,F_2} d_{\sigma_i}(F_1,F_2;H) \right). \end{aligned} \tag{3.3}$$

  The variables s_H are called *surplus* variables.

- Q_i is positive semi-definite for $i\in[r]$. (The matrices Q_i are often called the *block variables* of the SDP problem. We can assume without loss of generality that each Q_i is symmetric, as otherwise we could replace Q_i by $(Q_i+Q_i^T)/2$.)

A computer can solve this SDP problem numerically, allowing for an efficient determination of the inequalities required to prove the extremal problem. We note at this point, that the solution to the SDP problem need not only give the asymptotic bound, but can also provide some structural information about the extremal tree-pair.

4 Bounds on the SDP problem

In this section we discuss how we obtained the bounds for the main theorem and some other practical considerations relative to the main SDP problem. For a square matrix A, let $\mathrm{tr}(A)$ denote its trace. The original formulation of the SDP problem can be rewritten in concise matrix notation as follows:

$$\begin{array}{rrcll} \text{minimize} & \mathrm{tr}(C\cdot Q) & & & \\ \text{subject to} & \mathrm{tr}(A_j\cdot Q) & = & b_j, & \text{for } j=1,\ldots,m, \\ \text{and} & Q & \succeq & 0 & \end{array} \tag{4.4}$$

where $m=|\mathcal{F}_t|$ represents the number of constraints in the problem, C is the cost matrix (we have $\mathrm{tr}(C\cdot X)=y$, where y is as in Section 3.3, see e.g., (3.3)), $Q\succeq 0$ is positive semi-definite matrix consisting of all the block-variable matrices Q_i, and each equation $\mathrm{tr}(A_j\cdot Q)=b_j$ corresponds to one of the equations (3.3) from the original formulation. In particular, if $\mathcal{F}_t=\{H_1,\ldots,H_m\}$, we have $b_j=d(cr_4;H_j)$. Finally, we let ℓ denote the number of rows/columns of Q.

A computer usually cannot solve (4.4) exactly, but only approximately. In other words, the output of the SDP solver will be a matrix Q' that satisfies the constraints approximately. Namely, we have

$$\begin{array}{rll} \left|\mathrm{tr}(A_j\cdot Q')-b_j\right| & \leq \varepsilon, & \text{for } j=1,\ldots m, \text{ and} \\ Q'+\varepsilon I_\ell & \succeq 0, & \end{array} \tag{4.5}$$

for some small $\varepsilon>0$ (usually $\varepsilon<10^{-9}$), where I_ℓ denotes the $\ell\times\ell$ identity matrix. In what follows we describe how to obtain a matrix Q that satisfies all the constraints of (4.4) and is not "too far" from the approximate solution Q'. That way $\mathrm{tr}(C\cdot Q)\approx\mathrm{tr}(C\cdot Q')$.

A natural first step towards this goal is to slightly change Q' so that it satisfies all the linear constraints in (4.4). For that purpose, we will project Q' to the affine subspace of all $\ell\times\ell$ symmetric real matrices Q that satisfy $\mathrm{tr}(A_j\cdot Q)=b_j$ for all $j=1,\ldots,m$. Let Q'' denote this projection. How much did we change the approximate solution? Namely, how large is $||Q'-Q''||_\infty$? We recall that for a matrix A, we denote $||A||_\infty := \max_{ij}|A_{ij}|$ and $||A||_1 := \sum_{ij}|A_{ij}|$.

To estimate $||Q'-Q''||_\infty$ we often use some inequalities from the following proposition.

Proposition 4.1. *The following statements are true:*

(i) If $A\in\mathbb{R}^{m\times n}$ and $B\in\mathbb{R}^{n\times l}$ are two real matrices, then $||A\cdot B||_\infty \leq n\cdot||A||_\infty\cdot||B||_\infty$.

(ii) If $A\in\mathbb{R}^{m\times n}$ and $v\in\mathbb{R}^n$, then $||A\cdot v||_\infty \leq ||A||_\infty\cdot||v||_1$.

(iii) *Let $A \in \mathbb{R}^{n\times n}$ be any $n \times n$ symmetric matrix. Then $A + n \cdot ||A||_\infty \cdot I_n$ is positive semi-definite.*

Proof. (i) Let $C = A \cdot B$. We have $C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$, thus

$$\begin{aligned}|C_{ij}| \leq \sum_{k=1}^{n} |A_{ik}||B_{kj}| &\leq \sum_{k=1}^{n} ||A||_\infty ||B||_\infty \\ &= n \cdot ||A||_\infty \cdot ||B||_\infty,\end{aligned}$$

hence $||C||_\infty \leq n \cdot ||A||_\infty \cdot ||B||_\infty$.

(ii) Let $w = A \cdot v$. We have $w_i = \sum_{j=1}^{n} A_{ij}v_j$, thus

$$\begin{aligned}|w_i| \leq \sum_{j=1}^{n} |A_{ij}||v_j| &\leq \sum_{j=1}^{n} ||A||_\infty |v_j| \\ &= ||A||_\infty \cdot ||v||_1,\end{aligned}$$

therefore $||w||_\infty \leq ||A||_\infty \cdot ||v||_1$.

(iii) Let $B = A + n \cdot ||A||_\infty \cdot I_n$. It suffices to show that for all $v \in \mathbb{R}^n$, we have $v^T \cdot B \cdot v \geq 0$. Let $a = v^T \cdot A \cdot v$. Using the definition of B, we obtain

$$\begin{aligned}b &:= v^T \cdot B \cdot v = v^T \cdot A \cdot v + n \cdot ||A||_\infty \cdot ||v||_2^2 \\ &= a + n \cdot ||A||_\infty \cdot ||v||_2^2.\end{aligned}$$

By (ii) applied twice, we infer

$$\begin{aligned}|a| = ||v^T \cdot A \cdot v||_\infty &\leq ||v^T||_1 \cdot ||A \cdot v||_\infty \\ &\leq ||A||_\infty \cdot ||v||_1^2\end{aligned}$$

So by Cauchy-Schwarz inequality, we obtain $|a| \leq ||A||_\infty \cdot ||v||_1^2 \leq n \cdot ||A||_\infty \cdot ||v||_2^2$, therefore $b \geq 0$, finishing the proof.

In what follows, we introduce further notation in order to express Q'' in terms of Q' and the parameters of the problem (4.4). Let $\mathcal{S}$ be the linear space of all $\ell \times \ell$ real symmetric matrices, and let $\mathbf{A}$ be the linear map $\mathbf{A} : \mathcal{S} \to \mathbb{R}^m$ defined by $\mathbf{A}(Q)_j = \mathrm{tr}(A_j \cdot Q)$. In addition, let $b \in \mathbb{R}^m$ be vector with coordinates b_j for $j = 1, \ldots, m$, and let $\mathcal{H}$ be the affine subspace of all $\ell \times \ell$ real symmetric matrices Q that satisfy the linear constraints of (4.4), namely $\mathrm{tr}(A_j \cdot Q) = b_j$ for $j = 1, \ldots, m$. Note that $\mathcal{H}$ is the pre-image of b by $\mathbf{A}$. Let $\mathbf{P}$ be the orthogonal projection from the set $\mathcal{S}$ to the affine subspace $\mathcal{H}$. One can compute this projection by a solution to a least squares problem as follows:

$$\mathbf{P}(Q) = Q + \mathbf{A}^T \cdot (\mathbf{A} \cdot \mathbf{A}^T)^{-1}(b - \mathbf{A}(Q)),$$

for all $Q \in \mathcal{S}$, where $\mathbf{A}^T : \mathbb{R}^m \to \mathcal{S}$ denotes the transpose of $\mathbf{A}$. We have $Q'' = \mathbf{P}(Q')$, thus by Proposition 4.1 (i), we have

$$\begin{aligned}||Q'' - Q'||_\infty &\leq m \cdot ||\mathbf{A}^T \cdot (\mathbf{A} \cdot \mathbf{A}^T)^{-1}||_\infty \\ &\quad \cdot ||b - \mathbf{A}(Q')||_\infty \\ &\leq \varepsilon m \cdot ||\mathbf{A}^T \cdot (\mathbf{A} \cdot \mathbf{A}^T)^{-1}||_\infty.\end{aligned}$$

For all the instances of (4.4) that we consider, one can verify that $||\mathbf{A}^T \cdot (\mathbf{A} \cdot \mathbf{A}^T)^{-1}||_\infty \leq 1$, thus $\varepsilon' := ||Q'' - Q'||_\infty < \varepsilon m$. This inequality together with Proposition 4.1 (ii) implies that

$$\begin{aligned}\mathrm{tr}(C \cdot Q'') &= \mathrm{tr}(C \cdot Q') + \mathrm{tr}(C \cdot (Q'' - Q')) \\ &\leq \mathrm{tr}(C \cdot Q') + \varepsilon m \cdot ||C||_1.\end{aligned}$$

We know that Q'' satisfies all the linear constraints of (4.4), but Q'' might not be positive semi-definite. An application of Proposition 4.1 (iii) yields that

$$(Q'' - Q') + \ell \cdot ||Q'' - Q'||_\infty \cdot I_\ell \succeq 0,$$

which, together with the inequality $Q' + \varepsilon I_\ell \succeq 0$ from (4.5), implies that $Q'' + (\varepsilon + \ell\varepsilon') \cdot I_\ell \succeq 0$. To make Q'' positive semi-definite, we hope to find a matrix $\tilde{Q}$ that satisfies $\mathrm{tr}(A_j \cdot \tilde{Q}) = 0$ for $j = 1, \ldots, m$ and such that all the eigenvalues of $\tilde{Q}$ are large. If such $\tilde{Q}$ exists then $Q'' + \delta\tilde{Q}$ will be positive semi-definite for some small $\delta > 0$ and will also satisfy the linear constraints in (4.4). For this reason, we consider the following problem:

(4.6)

$$\begin{array}{rrcll} \text{minimize} & \mathrm{tr}(0 \cdot \tilde{Q}) & & & \\ \text{subject to} & \mathrm{tr}(A_j \cdot \tilde{Q}) & = & 0, & \text{for all } j = 1, \ldots, m, \\ \text{and} & \tilde{Q} & \succ & 0, & \end{array}$$

where $\tilde{Q} \succ 0$ is strictly positive-definite.

Note that the function being minimized is the constant zero function, so (4.6) is a pure feasibility problem. We again use computers to obtain an approximate solution $\tilde{Q}'$ to (4.6). Surprisingly, it turns out that the obtained solution $\tilde{Q}'$ not only satisfies $\left|\mathrm{tr}(A_j \cdot \tilde{Q}')\right| < \varepsilon$ for all $j = 1, \ldots, m$, but also has a large smallest eigenvalue (much larger than $\varepsilon + \ell\varepsilon'$), even though $|\mathrm{tr}(C \cdot \tilde{Q}')|$ is relatively small. We will later exploit these properties to adjust Q'' to an exact solution of (4.4).

Using similar ideas as before, we obtain a matrix $\tilde{Q}''$ that satisfies $\mathrm{tr}(A_j \cdot \tilde{Q}'') = 0$ for all $j = 1, \ldots, m$ by means of orthogonal projection of $\tilde{Q}'$ to the appropriate subspace. As we have already seen, this operation only slightly changes the eigenvalues of $\tilde{Q}'$.

Finally we let $Q := Q'' + \delta\tilde{Q}''$, where $\delta = \frac{\varepsilon+\ell\varepsilon'}{\lambda}$ and λ is the smallest eigenvalue of $\tilde{Q}''$ (in all of our instances we have $\delta < 10^{-4}$). Clearly Q satisfy all the constraints of (4.4), including $Q \succeq 0$. Moreover, we have

$$\begin{aligned}\operatorname{tr}(C \cdot Q) &= \operatorname{tr}(C \cdot Q'') + \delta \cdot \operatorname{tr}(C \cdot \tilde{Q}'') \\ &\leq \operatorname{tr}(C \cdot Q') + \varepsilon m \cdot ||C||_1 + \delta \cdot \operatorname{tr}(C \cdot \tilde{Q}''),\end{aligned}$$

and since both εm and δ are typically small, we will not change the objective value much from the original approximate solution Q' to the exact solution Q. Therefore Q is the desired exact solution which is "close" to Q'.

In what follows we have a compiled table displaying the several bounds obtained for different instances of the SDP problem. he first column represents the parameter t, which is the size of the tree-pairs used in the expansion of the problem (see Section 3.3 for more details). The second column counts the number of tree-pairs of size t. The third column indicates how many types where used, that is, the types σ for the inequalities of the form (3.2). The used types are all those having size with the same parity as and strictly smaller than t. The fourth column contains the total number of variables in the SDP instance including the surpluses. Finally, the last column tells the bound obtained from the SDP solver. The program used to generate the SDP instances and verify these calculations can be downloaded at `http://www.ima.umn.edu/~hnaves/papers/quartet.zip`.

t	$m = \vert\mathcal{F}_t\vert$	# of types	ℓ	Bound
5	4	1	50	0.884766
6	31	3	697	0.760257
7	243	6	12050	0.707633
8	3532	35	506171	0.688397

Table 1: Several instances of the main SDP problem.

5 On caterpillar trees

In this section, we prove Theorem 1.2 — a restricted version of Conjecture 1.1 to caterpillar trees. One possible approach to this problem is to use the same machinery of flag algebras for the theory of tree-pairs restricted to caterpillar trees, and try to obtain a bound in the same way as we did for Theorem 1.1. However, this approach does not immediately yield the bound of $\frac{2}{3}$, and so it is necessary (and worthwhile) to think about this problem from a different perspective. In the next few paragraphs we will explain how to map the problem of computing the induced density of cr_4 in a tree-pair of caterpillar trees into a problem of counting induced sub-permutations of size 4.

Suppose $D = \{\overline{T}_1, \overline{T}_2\}$ is a tree-pair composed by two caterpillar trees on $n+2$ leaves (as exemplified in Figure 2). One can think of D as a permutation of $\{\alpha, x_1, \ldots, x_n, \beta\}$ — a permutation that tells us exactly how the leaves of T_1 are "attached" to the leaves of T_2. For instance, the tree-pair cr_5^A in Figure 10 could be represented by the permutation $\alpha \to \alpha, x_1 \to x_1, x_2 \to x_3, x_3 \to x_2, \beta \to \beta$. In fact, multiple permutations might give rise to the same tree-pair. Regarding this matter, our first observation is that any caterpillar tree on four or more leaves has exactly 8 distinct automorphisms. To illustrate this observation, consider the caterpillar tree on $n+2$ leaves labelled by $\alpha, x_1, \ldots, x_n, \beta$ as depicted in Figure 11. One of the automorphisms of this tree is σ_1, which is the unique automorphism that maps α to β and β back to α, such as in a "reflection". Similarly, σ_2 is the automorphism that only swaps α with x_1 and leaves all the remaining vertices in place. Finally, σ_3 is the automorphism that swaps β and x_n. The group of automorphisms can be then written as $\{\sigma_1^{i_1}\sigma_2^{i_2}\sigma_3^{i_3} : 0 \leq i_1, i_2, i_3 \leq 1\}$. Our second observation is that given a permutation π, and two automorphisms σ, σ' of the caterpillar tree with leaves labelled $\{\alpha, x_1, \ldots, x_n, \beta\}$, the permutation $\sigma\pi\sigma'$ represents the same tree-pair as π itself. Here we think of σ and σ' as only acting solely on the leaves of the caterpillar trees.

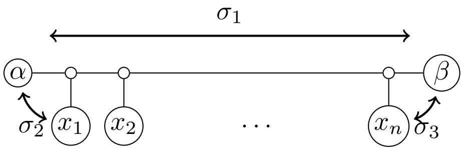

Figure 11: The automorphisms of a caterpillar tree.

Given a permutation π of $L := \{\alpha, x_1, \ldots, x_n, \beta\}$, how do we count the number of induced copies of cr_4 in the corresponding tree-pair D represented by π? Suppose $S \subseteq L$ is a subset of size 4 such that $\alpha, \beta \notin S$ and $\alpha, \beta \notin \pi(S)$, say $S = \{x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}\}$ with $\pi(x_{i_t}) = x_{j_t}$ for $t = 1, \ldots, 4$ and $i_1 < i_2 < i_3 < i_4$. Here it is helpful to think of S as a subset of the leaves of T_1 before the identification with the leaves of T_2. The corresponding leaves selected by S will induce a copy of id_4 in D if either $\max\{j_1, j_2\} < \min\{j_3, j_4\}$, or $\max\{j_3, j_4\} < \min\{j_1, j_2\}$. Otherwise S induces a copy of cr_4. Since there are only $O(n^3)$ subsets S that do not satisfy the condition $\{\alpha, \beta\} \cap (S \cup \pi(S)) = \emptyset$, the problem of computing the density of id_4 in D essentially becomes the problem of computing the density of the following induced sub-permutations in a permutation $\pi \in S_n$:

$$1234, 1243, 2134, 2143, 3412, 4312, 3421, 4321. \tag{5.7}$$

The machinery of flag algebras is very general and thus can also be applied to the theory of permutations. In fact, we have the following theorem which implies Theorem 1.2.

Theorem 5.1. *The sum of the densities of the permutations listed in* (5.7) *inside a permutation* $\pi \in S_n$ *is at least* $\frac{1}{3} + o(1)$ *for* n *large.*

Proof. Using the notation from flag algebras, let ϕ denote the sum of the densities of the permutations in (5.7), that is,

$$\begin{aligned}\phi = {} & 1234 + 1243 + 2134 + 2143 \\ & + 3412 + 4312 + 3421 + 4321.\end{aligned}$$

In this notation, a flag is just a permutation with some entries labeled by the set $[k]$ for some $k \geq 0$. For instance $1\textcircled{6}_2 42\textcircled{3}_1 5$ denotes a flag whose underlying permutation is 164235 for which the fifth entry is labeled 1 and the second entry is labeled 2. In this case, the type of the flag is $\textcircled{2}_2\textcircled{1}_1$, since the sub-permutation induced by the labeled entries is isomorphic to 21 (corresponding to the entries 63). Another example is the flag $\textcircled{5}_3 17\textcircled{2}_1 3\textcircled{6}_2 4$ — its underlying permutation is 5172364 and its type is $\textcircled{2}_3\textcircled{1}_1\textcircled{3}_2$. With this definitions in mind, we remark that a type is just a permutation of $[k]$ with all entries labeled by elements of the set $[k]$. Thus, for an integer $k \geq 0$, types on k entries are in one-to-one correspondence with pairs of permutations of $[k]$.

Consider the following 4 types of size 2

$$\begin{array}{rclcrcl} \rho_1 & = & \textcircled{1}_1\textcircled{2}_2, & \quad & \rho_2 & = & \textcircled{2}_1\textcircled{1}_2, \\ \rho_3 & = & \textcircled{1}_2\textcircled{2}_1, & & \rho_4 & = & \textcircled{2}_2\textcircled{1}_1. \end{array}$$

We have

$$\phi = \frac{1}{3} + \sum_{i=1}^{4} 3\cdot[[(X_i - Y_i)^2]]_{\rho_i} + 6\cdot[[(X_i - Z_i)^2]]_{\rho_i} \tag{5.8}$$

where

$$\begin{aligned}
X_1 &= -1\textcircled{2}_1\textcircled{3}_2 4 - 1\textcircled{2}_1\textcircled{4}_2 3 + 4\textcircled{2}_1\textcircled{3}_2 1 + 3\textcircled{2}_1\textcircled{4}_2 1, \\
Y_1 &= +1\textcircled{2}_1 3\textcircled{4}_2 + 1\textcircled{2}_1 4\textcircled{3}_2 - 4\textcircled{2}_1 1\textcircled{3}_2 - 3\textcircled{2}_1 1\textcircled{4}_2, \\
Z_1 &= +\textcircled{2}_1 1\textcircled{3}_2 4 + \textcircled{2}_1 1\textcircled{4}_2 3 - \textcircled{2}_1 4\textcircled{3}_2 1 - \textcircled{2}_1 3\textcircled{4}_2 1, \\
X_2 &= -4\textcircled{3}_1\textcircled{2}_2 1 - 4\textcircled{3}_1\textcircled{1}_2 2 + 1\textcircled{3}_1\textcircled{2}_2 4 + 2\textcircled{3}_1\textcircled{1}_2 4, \\
Y_2 &= +4\textcircled{3}_1 2\textcircled{1}_2 + 4\textcircled{3}_1 1\textcircled{2}_2 - 1\textcircled{3}_1 4\textcircled{2}_2 - 2\textcircled{3}_1 4\textcircled{1}_2, \\
Z_2 &= +\textcircled{3}_1 4\textcircled{2}_2 1 + \textcircled{3}_1 4\textcircled{1}_2 2 - \textcircled{3}_1 1\textcircled{2}_2 4 - \textcircled{3}_1 2\textcircled{1}_2 4, \\
X_3 &= -1\textcircled{2}_2\textcircled{3}_1 4 - 2\textcircled{1}_2\textcircled{3}_1 4 + 4\textcircled{2}_2\textcircled{3}_1 1 + 4\textcircled{1}_2\textcircled{3}_1 2, \\
Y_3 &= +\textcircled{1}_2 2\textcircled{3}_1 4 + \textcircled{2}_2 1\textcircled{3}_1 4 - \textcircled{2}_2 4\textcircled{3}_1 1 - \textcircled{1}_2 4\textcircled{3}_1 2, \\
Z_3 &= +1\textcircled{2}_2 4\textcircled{3}_1 + 2\textcircled{1}_2 4\textcircled{3}_1 - 4\textcircled{2}_2 1\textcircled{3}_1 - 4\textcircled{1}_2 2\textcircled{3}_1, \\
X_4 &= -4\textcircled{3}_2\textcircled{2}_1 1 - 3\textcircled{4}_2\textcircled{2}_1 1 + 1\textcircled{3}_2\textcircled{2}_1 4 + 1\textcircled{4}_2\textcircled{2}_1 3, \\
Y_4 &= +\textcircled{4}_2 3\textcircled{2}_1 1 + \textcircled{3}_2 4\textcircled{2}_1 1 - \textcircled{3}_2 1\textcircled{2}_1 4 - \textcircled{4}_2 1\textcircled{2}_1 3, \\
Z_4 &= +4\textcircled{3}_2 1\textcircled{2}_1 + 3\textcircled{4}_2 1\textcircled{2}_1 - 1\textcircled{3}_2 4\textcircled{2}_1 - 1\textcircled{4}_2 3\textcircled{2}_1,
\end{aligned}$$

therefore $\phi \geq \frac{1}{3}$, thereby proving the theorem. Note that in order to attest the correctness of (5.8), it suffices to evaluate the left- and the right-hand side of the equation for all permutations of size 6.

6 Concluding remarks

In Theorem 1.1 we showed that the maximum quartet distance between two arbitrary phylogenetic trees on n leaves is at most $(0.69 + o(1))\binom{n}{4}$. It would be interesting to know if the techniques of this paper can be pushed even further to obtain the $(\frac{2}{3} + o(1))\binom{n}{4}$ thereby establishing Conjecture 1.1.

Another approach to Conjecture 1.1 is to solve an extremal problem in the theory of 4-uniform hypergraphs. In [1], Alon *et al* proved the asymptotic upper bound of $\frac{9}{10}\binom{n}{4}$ by mapping a tree-pair into a 4-uniform hypergraph in the following way. The vertices of the hypergraph are the leaves of the tree-pair and a subset S of 4 leaves is an edge of the hypergraph if the sub-tree-pair induced by S is isomorphic to cr_4. They showed that the resulting hypergraph $\mathcal{H}$ does not contain a copy of K_6^4 — the complete 4-uniform hypergraph on 6 vertices. One remark is that not only K_6^4 but also several other forbidden hypergraphs do not appear as induced subgraphs of $\mathcal{H}$. A natural question emerges: can one characterize this family of forbidden subgraphs? In particular, is it finite?

Acknowledgement We thank Sagi Snir and Raphy Yuster for helpful discussions.

References

[1] N. Alon, S. Snir, and R. Yuster, *On the compatibility of quartet trees*, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA (2014), 535–545.

[2] R. Baber, and J. Talbot, *Hypergraphs do jump*, Combinatorics, Probability and Computing **20** 2 (2011), 161–171.

[3] J. Balogh, P. Hu, B. Lidický, O. Pikhurko, B. Udvari, and J. Volec, *Minimum Number of Monotone Subsequences of Length 4 in Permutations*, Combinatorics, Probability and Computing, to appear.

[4] H. Bandelt, and A. Dress, *Reconstructing the shape of a tree from observed dissimilarity data*, Advances in Applied Mathematics **7** (1986), 309–343.

[5] V. Berry, and O. Gascuel, *Inferring evolutionary trees with strong combinatorial evidence*, Theoretical Computer Science **240** (2001), 271–298.

[6] V. Berry, T. Jiang, P. Kearney, M. Li, and T. Wareham, *Quartet cleaning: improved algorithms and simulations*, European Symposium on Algorithms (1999).

[7] H. Colonius, and H. Schulze, *Tree structures for proximity data*, British Journal of Mathematical and Statistical Psycology **34(2)** (1981), 167–180.

[8] S. Das, H. Huang, J. Ma, H. Naves, and B. Sudakov, *A problem of Erdős on the minimum number of k-cliques*, Journal of Combinatorial Theory Series B **103** (2013), 344–373.

[9] G. Estabrook, F. McMorris, and C. Meacham, *Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units*, Systematic Biology **34(2)** (1985), 193–200.

[10] R. Glebov, D. Král, J. Volec, *An application of flag algebras to a problem of Erdős and Sós*, Electronic Notes in Discrete Mathematics **43** (2013), 171–177.

[11] H. Hatami, J. Hladký, D. Král', S. Norine, and A. Razborov, *On the number of pentagons in triangle-free graphs*, Journal of Combinatorial Theory Series A **120** (2013), 722–732.

[12] T. Jiang, P. Kearney, and M. Li, *Orchestrating quartets: approximation and data correlation*, IEEE Symposium Foundation of Computer Science (FOCS), pages 416–425, Palo Alto, California, November 1998.

[13] K. St. John, T. Warnow, B. Moret, and L. Vawter, *Performance study of phylogenetic methods: (unweighted quartet methods and neighbor-joining*, Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (2001).

[14] P. Keevash, *Hypergraph Turán Problems*, Surveys in combinatorics, Cambridge (2011).

[15] D. R. Maddison, and K.-S. Schulz (eds.) 2007. *The Tree of Life Web Project*. Internet address: `http://tolweb.org`.

[16] O. Pikhurko, and E. R. Vaughan, *Minimum Number of k-Cliques in Graphs with Bounded Independence Number*, Combinatorics Probability and Computing **22** (2013), 910–934.

[17] V. Falgas-Ravry, and E. R. Vaughan, *Applications of the semi-definite method to the Turán density problem for 3-graphs*, Combinatorics Probability and Computing **22** (2013), 21–54.

[18] A. Razborov, *Flag algebras*, Journal of Symbolic Logic **72(4)** (2007), 1239–1282.

[19] A. Razborov, *On 3-hypergraphs with forbidden 4-vertex configurations*, SIAM Journal of Discrete Mathematics **24** (2010), 946–963.

[20] A. Razborov, *On the minimum density of triangles in graphs*, Combinatorics Probability and Computing **17(4)** (2008), 603–618.

[21] C. Semple, and M. A. Steel, **Phylogenetics**, Oxford University Press (2003).

[22] S. Snir, and S. Rao, *Quartets maxcut: A divide and conquer quartets algorithm*, Transactions on Computational Biology and Bioinformatics (TCBB) **7(4)** (2010), 714–718.

[23] S. Snir, and R. Yuster, *Reconstructing approximate phylogenetic trees from quartet samples*, SIAM Journal on Computing **41(6)** (2012), 1466–1480.

[24] M. Steel, *The complexity of reconstructing trees from qualitative characters and subtrees*, Journal of Classification **9(1)** (1992), 91–116.

[25] K. Strimmer, and A. von Haeseler, *Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies*, Molecular Biology and Evolution **13(7)** (1996), 964–969. Software available at `ftp://ftp.ebi.ac.uk/pub/software/unix/puzzle/`.